

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
Khoa Công nghệ Thông tin



CSC14003 – Cơ sở trí tuệ nhân tạo

Report

Decision Tree Classification

21127021 - Trương Văn Chí
21CLC06

Giảng viên hướng dẫn

Nguyễn Ngọc Thảo

Lê Ngọc Thành

Nguyễn Trần Duy Minh

Hồ Chí Minh - 07/2023

MỤC LỤC

DANH MỤC ẢNH	i
DANH MỤC BẢNG	ii
1 Tỷ lệ hoàn thành công việc	1
2 Chuẩn bị tập dữ liệu	1
2.1 Đọc dữ liệu mẫu	1
2.2 Chia cắt dữ liệu	2
2.3 Trực quan hóa tập dữ liệu	3
3 Xây dựng mô hình Decision Tree	6
3.1 Mô hình tỷ lệ 40/60	6
3.2 Mô hình tỷ lệ 60/40	6
3.3 Mô hình tỷ lệ 80/20	6
3.4 Mô hình tỷ lệ 90/10	6
4 Đánh giá kết quả	6
4.1 Kết quả cho tỷ lệ 40/60	7
4.2 Kết quả cho tỷ lệ 60/40	8
4.3 Kết quả cho tỷ lệ 80/20	10
4.4 Kết quả cho tỷ lệ 90/10	11
4.5 Nhận xét tổng quan	12
5 Độ sâu và độ chính xác của mô hình	13
5.1 Trực quan hóa đồ thị	13
5.2 Bảng thống kê	15
5.3 Đồ thị trực quan hóa	15

5.4	Nhận xét	15
-----	--------------------	----

DANH MỤC HÌNH ẢNH

Hình 2.1	Thông tin chi tiết về tập dữ liệu	2
Hình 2.2	File dữ liệu sau khi được chia	3
Hình 2.3	Trực quan hóa dữ liệu tỉ lệ 40/60	4
Hình 2.4	Trực quan hóa dữ liệu tỉ lệ 60/40	4
Hình 2.5	Trực quan hóa dữ liệu tỉ lệ 80/20	5
Hình 2.6	Trực quan hóa dữ liệu tỉ lệ 90/10	5
Hình 4.7	Classifier report cho tỉ lệ 40/60	7
Hình 4.8	Confusion matrix cho tỉ lệ 40/60	8
Hình 4.9	Classifier report cho tỉ lệ 60/40	8
Hình 4.10	Confusion matrix cho tỉ lệ 60/40	9
Hình 4.11	Classifier report cho tỉ lệ 80/20	10
Hình 4.12	Confusion matrix cho tỉ lệ 80/20	11
Hình 4.13	Classifier report cho tỉ lệ 90/10	11
Hình 4.14	Confusion matrix cho tỉ lệ 90/10	12
Hình 5.15	Mô hình cây với max_depth = 2	13
Hình 5.16	Mô hình cây với max_depth = 3	14
Hình 5.17	Mô hình cây với max_depth = 4	14
Hình 5.18	Đồ thị ảnh hưởng giữa độ sâu và độ sai sót	15

DANH MỤC BẢNG BIỂU

Bảng 1.1	Bảng biểu tỉ lệ hoàn thành công việc	1
Bảng 5.2	Bảng thống kê ảnh hưởng của độ sâu đến độ sai lệch	15

1 Tỉ lệ hoàn thành công việc

Bảng 1.1 Bảng biểu tỉ lệ hoàn thành công việc

No.	Specifications	Score(%)	Complete (%)
1	Preparing the data sets	30	100
2	Building the decision tree classifiers	20	100
3	Evaluating the decision tree classifiers		
	Classification report and confusion matrix	10	100
	Comments	10	100
4	The depth and accuracy of a decision tree		
	Trees, tables, and charts	20	100
	Comments	10	100
Total		100	100

2 Chuẩn bị tập dữ liệu

2.1 Đọc dữ liệu mẫu

Ta dùng thư viện **pandas** và dùng hàm **pandas.read_csv()** để đọc dữ liệu từ file **./nursery/nursery.data.csv** vào biến **df**

Ta thấy được tập dữ liệu ta gồm 9 cột, trong đó 8 cột đầu chứa các đặc trưng và cột thứ 9 là nhãn của đặc trưng tương ứng.

Ta gán tên cho từng cột tương ứng và dùng hàm **df.info()** để xem rõ hơn về dữ liệu.

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12959 entries, 0 to 12958
Data columns (total 9 columns):
#   Column      Non-Null Count  Dtype
---  -
0   parents     12959 non-null  object
1   has_nurs    12959 non-null  object
2   form        12959 non-null  object
3   children    12959 non-null  object
4   housing     12959 non-null  object
5   finance     12959 non-null  object
6   social      12959 non-null  object
7   health      12959 non-null  object
8   class       12959 non-null  object
dtypes: object(9)
memory usage: 911.3+ KB

```

Hình 2.1 Thông tin chi tiết về tập dữ liệu

Như hình ảnh trên, ta có thể thấy rõ tập dữ liệu bao gồm 12959 hàng và 9 cột, các phần tử đều ở kiểu dữ liệu Object và không có phần tử nào thiếu sót thông tin

















2.2 Chia cắt dữ liệu

Từ dữ liệu mẫu ban đầu, ta chia dữ liệu thành các thành phần với tỉ lệ mong muốn. Ban đầu ta chia tập dữ liệu thành 2 tập nhỏ hơn là tập **Train** và tập **Test**. Ta xáo trộn dữ liệu trong df bằng hàm lấy mẫu **df.sample(frac = 1)**, hàm này sẽ trả về cho ta một mẫu ngẫu nhiên được lấy từ **df** với tỉ lệ frac (ở đây cụ thể là 1) so với tập dữ liệu ban đầu.

Ta chia tập dữ liệu ra 2 tập nhỏ hơn bằng toán tử split trong pandas

df.iloc[<hàng bắt đầu> : <hàng kết thúc trí kết thúc>, <cột bắt đầu> : <cột kết thúc>]

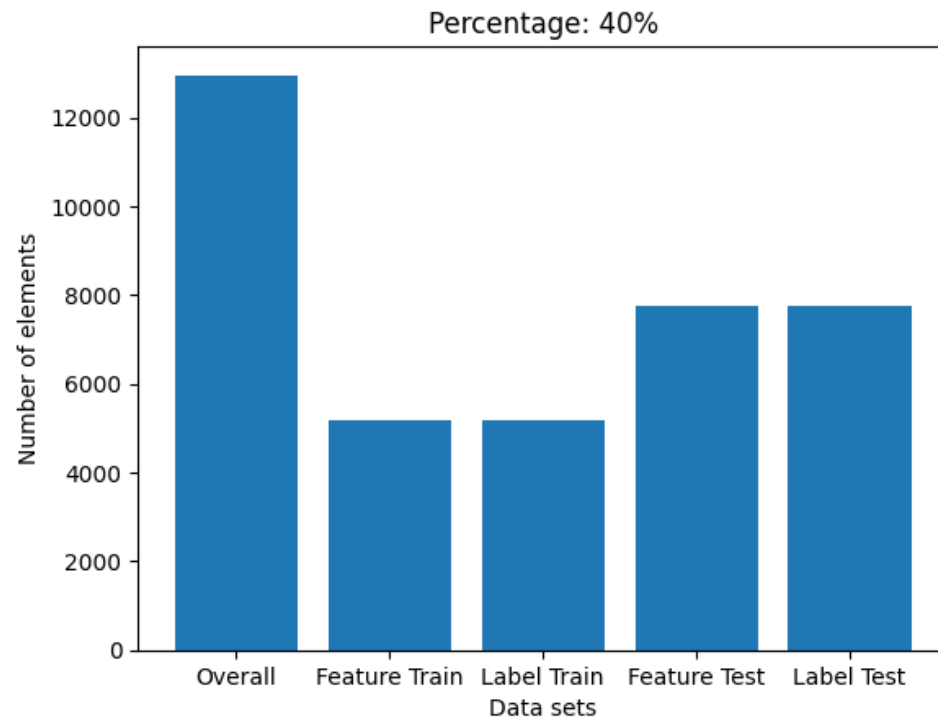
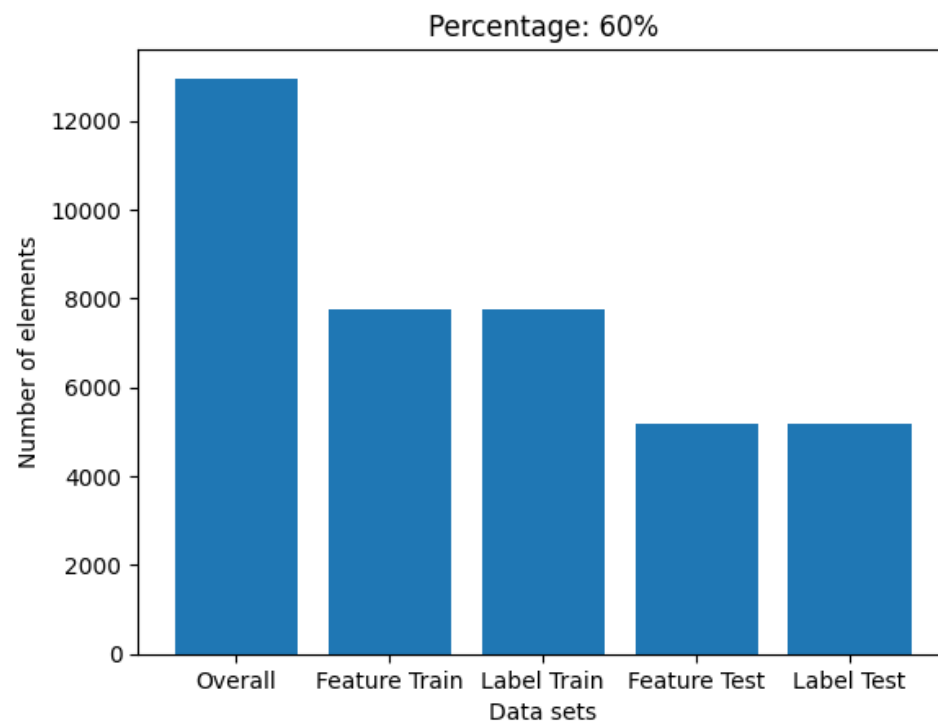
Sau khi có được các tập dữ liệu với tỉ lệ tương ứng ta ghi các tập này vào file để tiện cho việc xử lý các tác vụ phía sau. Tất cả tập dữ liệu được lưu trong folder **sample**.

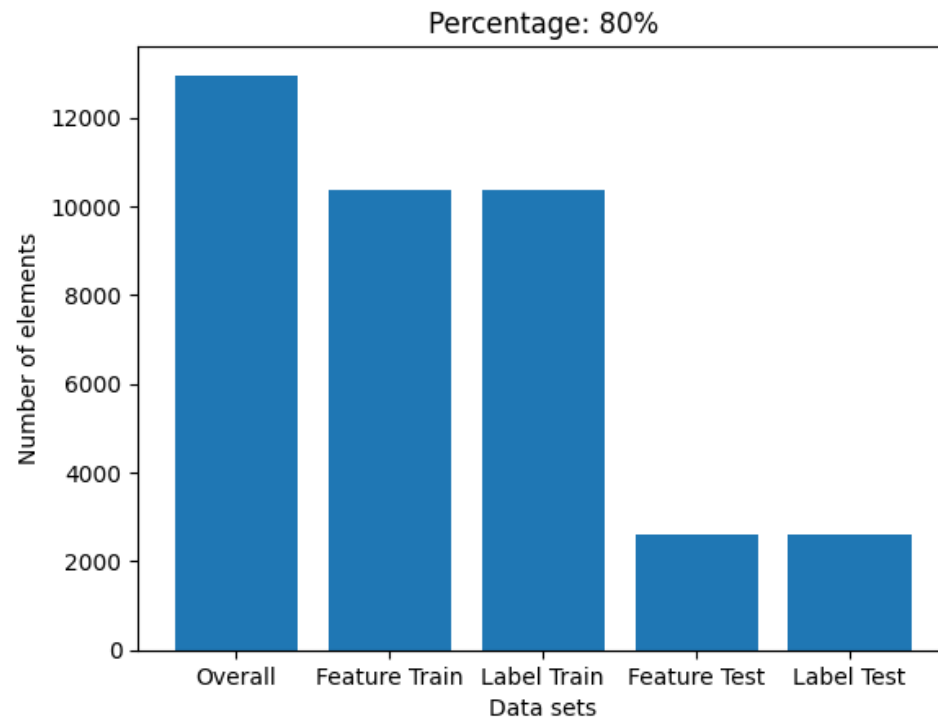
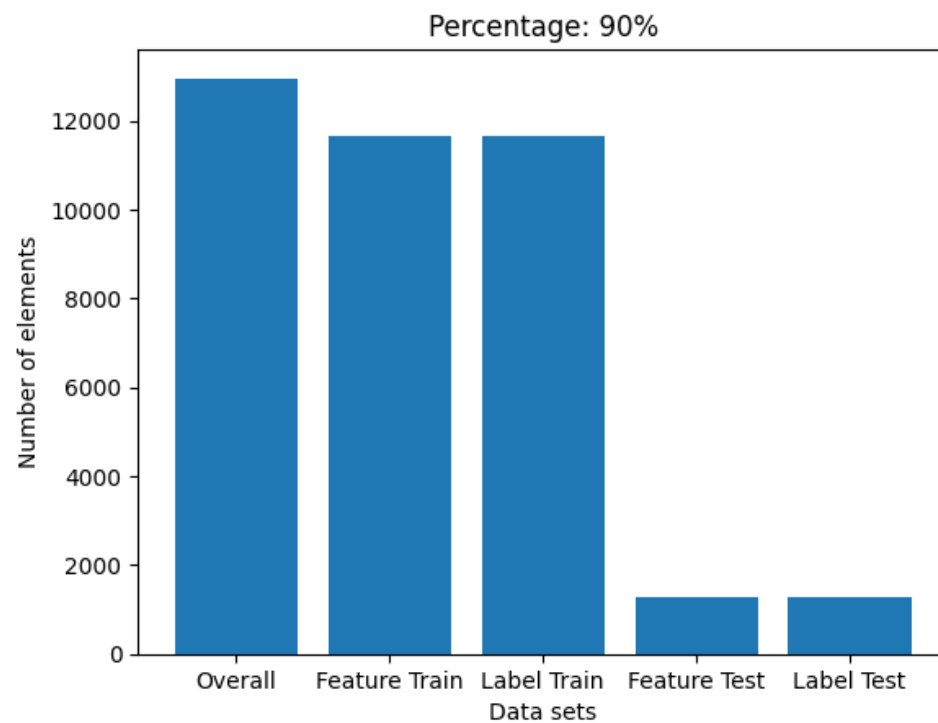
 40_feature_test.csv	8/17/2023 4:31 PM	Microsoft Excel Co...	553 KB
 40_feature_train.csv	8/17/2023 4:31 PM	Microsoft Excel Co...	368 KB
 40_label_test.csv	8/17/2023 4:31 PM	Microsoft Excel Co...	84 KB
 40_label_train.csv	8/17/2023 4:31 PM	Microsoft Excel Co...	56 KB
 60_feature_test.csv	8/17/2023 4:31 PM	Microsoft Excel Co...	369 KB
 60_feature_train.csv	8/17/2023 4:31 PM	Microsoft Excel Co...	552 KB
 60_label_test.csv	8/17/2023 4:31 PM	Microsoft Excel Co...	56 KB
 60_label_train.csv	8/17/2023 4:31 PM	Microsoft Excel Co...	84 KB
 80_feature_test.csv	8/17/2023 4:31 PM	Microsoft Excel Co...	185 KB
 80_feature_train.csv	8/17/2023 4:31 PM	Microsoft Excel Co...	737 KB
 80_label_test.csv	8/17/2023 4:31 PM	Microsoft Excel Co...	28 KB
 80_label_train.csv	8/17/2023 4:31 PM	Microsoft Excel Co...	112 KB
 90_feature_test.csv	8/17/2023 4:31 PM	Microsoft Excel Co...	92 KB
 90_feature_train.csv	8/17/2023 4:31 PM	Microsoft Excel Co...	829 KB
 90_label_test.csv	8/17/2023 4:31 PM	Microsoft Excel Co...	14 KB
 90_label_train.csv	8/17/2023 4:31 PM	Microsoft Excel Co...	126 KB

Hình 2.2 File dữ liệu sau khi được chia

2.3 Trực quan hóa tập dữ liệu

Ta dùng thư viện `matplotlib.pyplot.bar()` để trực quan hóa các tập dữ liệu. Cột **Overall** biểu diễn tổng số phần tử của dữ liệu, theo sau đó là các cột tương ứng biểu diễn số phần tử cho từng phần dữ liệu.

**Hình 2.3** Trực quan hóa dữ liệu tỉ lệ 40/60**Hình 2.4** Trực quan hóa dữ liệu tỉ lệ 60/40

**Hình 2.5** Trực quan hóa dữ liệu tỉ lệ 80/20**Hình 2.6** Trực quan hóa dữ liệu tỉ lệ 90/10

3 Xây dựng mô hình Decision Tree

Thư viện `sklearn.tree.DecisionTreeClassifier()` chỉ chạy được trên bộ dữ liệu là số. Dữ liệu từ tập đầu vào của ta đang ở định dạng Object (tức là category) nên ta cần phải có thêm một bước tiền xử lý dữ liệu để có thể chạy được mô hình này. Ta dùng hàm `sklearn.preprocessing.OrdinalEncoder()` để chuyển đổi dữ liệu trong tập `feature_train` từ category trở thành số nguyên.

Ta xuất cây kết quả dưới dạng đồ thị qua phương thức `sklearn.tree.export_graphviz` và dùng kết quả này để trực quan hóa cây kết quả dưới dạng hình ảnh thông qua thư viện `graphviz.Source` và in ra màn hình với `IPython.display.display()`

3.1 Mô hình tỉ lệ 40/60

Xem trực quan hóa cây kết quả tại `./Source/graph/40_sample.png` hoặc trong file `21127021.ipynb`

3.2 Mô hình tỉ lệ 60/40

Xem trực quan hóa cây kết quả tại `./Source/graph/60_sample.png` hoặc trong file `21127021.ipynb`

3.3 Mô hình tỉ lệ 80/20

Xem trực quan hóa cây kết quả tại `./Source/graph/80_sample.png` hoặc trong file `21127021.ipynb`

3.4 Mô hình tỉ lệ 90/10

Xem trực quan hóa cây kết quả tại `./Source/graph/90_sample.png` hoặc trong file `21127021.ipynb`

4 Đánh giá kết quả

Sau khi có kết quả chạy mô hình từ phần 3, ta tạo classifiers report và confusion matrix cho từng mẫu tỉ lệ.

Ta tạo ra label dự đoán `y_label` được sinh ra từ tập `feature_test` và mô hình ở phần 3 bằng phương thức `tree.predict()`.

Sau khi có được phần kết quả dự đoán của mô hình, ta kết hợp với kết quả chính xác của mô hình qua tập dữ liệu `label_test` và sử dụng hàm `sklearn.metrics.classification_report()` và `sklearn.metrics.confusion_matrix()` để lần lượt tính toán kết quả cho classifiers report

và confusion matrix.

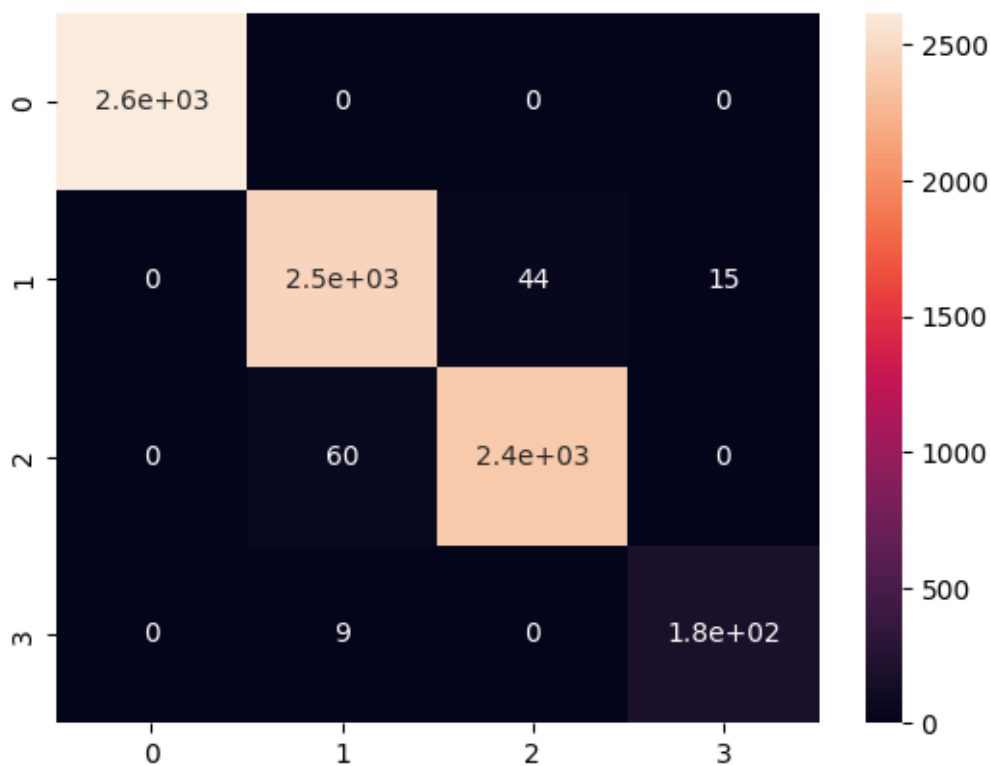
4.1 Kết quả cho tỉ lệ 40/60

	precision	recall	f1-score	support
not_recom	1.00	1.00	1.00	2618
priority	0.97	0.98	0.97	2522
recommend	0.00	0.00	0.00	1
spec_prior	0.98	0.98	0.98	2443
very_recom	0.92	0.95	0.94	190
accuracy			0.98	7774
macro avg	0.77	0.78	0.78	7774
weighted avg	0.98	0.98	0.98	7774

Hình 4.7 Classifier report cho tỉ lệ 40/60

Với classifier report phía trên ta đưa ra được nhận xét rằng:

- Với lớp not_recom, mô hình dự đoán đúng hoàn toàn với hệ số f1-score là 1.
- Lớp priority và spec_prior cũng được dự đoán với độ chính xác khá cao với hệ số f1-score lần lượt là 0.97 và 0.98
- Lớp very_recom được dự đoán sai khá nhiều với hệ số f1-score chỉ có 0.94
- Lớp recommend có hệ số f1-score bằng 0 vì ta không có đủ dữ liệu để train và test cho lớp này (tổng số 2 trên 12959)



Hình 4.8 Confusion matrix cho tỉ lệ 40/60

Theo như confusion matrix phía trên ta thấy chủ yếu các lớp được gán nhãn nhầm giữa nhãn 2 và 1. Cũng có một số ít nhãn được gán nhầm giữa 3 và 1 và không có sự nhầm lẫn của các nhãn khác.

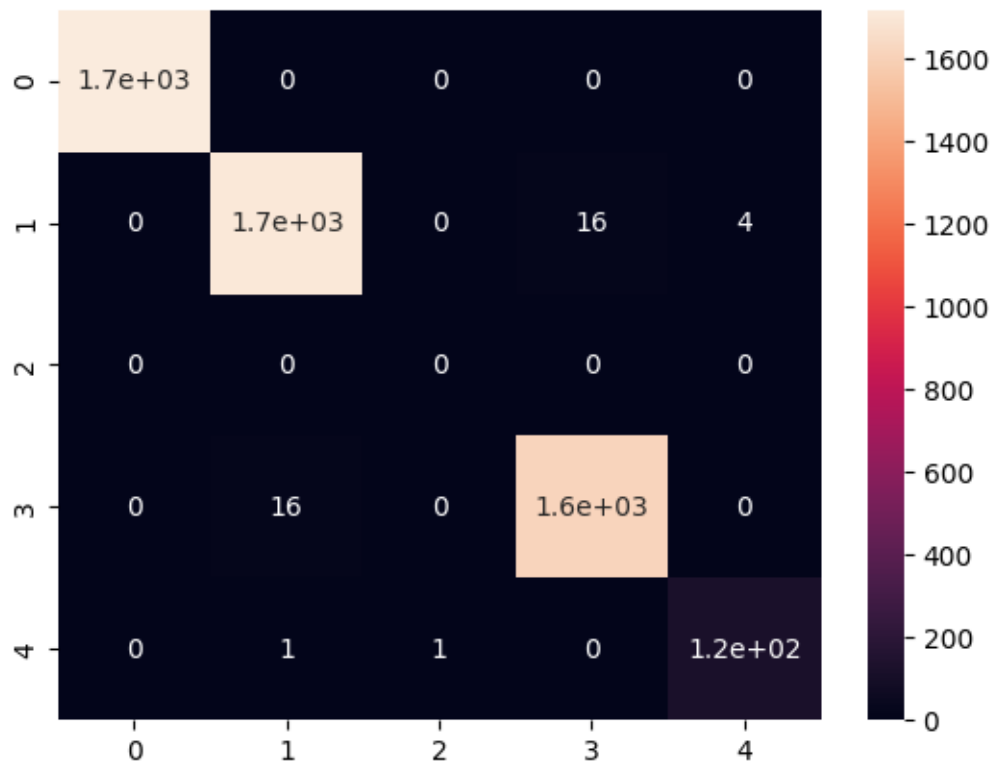
4.2 Kết quả cho tỉ lệ 60/40

	precision	recall	f1-score	support
not_recom	1.00	1.00	1.00	1717
priority	0.99	0.99	0.99	1719
recommend	0.00	0.00	0.00	0
spec_prior	0.99	0.99	0.99	1630
very_recom	0.97	0.98	0.97	117
accuracy			0.99	5183
macro avg	0.79	0.79	0.79	5183
weighted avg	0.99	0.99	0.99	5183

Hình 4.9 Classifier report cho tỉ lệ 60/40

Với classifier report phía trên ta đưa ra được nhận xét rằng:

- Với lớp not_recom, mô hình dự đoán đúng hoàn toàn với hệ số f1-score là 1.
- Lớp priority và spec_prior cũng được dự đoán với độ chính xác gần như tuyệt đối với hệ số f1-score cùng bằng 0.99
- Lớp very_recom được dự đoán sai nhiều nhất nhưng vẫn ở mức cao với hệ số f1-score là 0.97
- Lớp recommend có hệ số f1-score bằng 0 vì ta không có đủ dữ liệu để train và test cho lớp này (tổng số 2 trên 12959)



Hình 4.10 Confusion matrix cho tỉ lệ 60/40

Theo như confusion matrix phía trên ta thấy chủ yếu các lớp được gán nhãn nhầm giữa nhãn 3 và 1. Cũng có một số rất ít các nhãn được gán nhầm ở các lớp 1, 2 và 4.

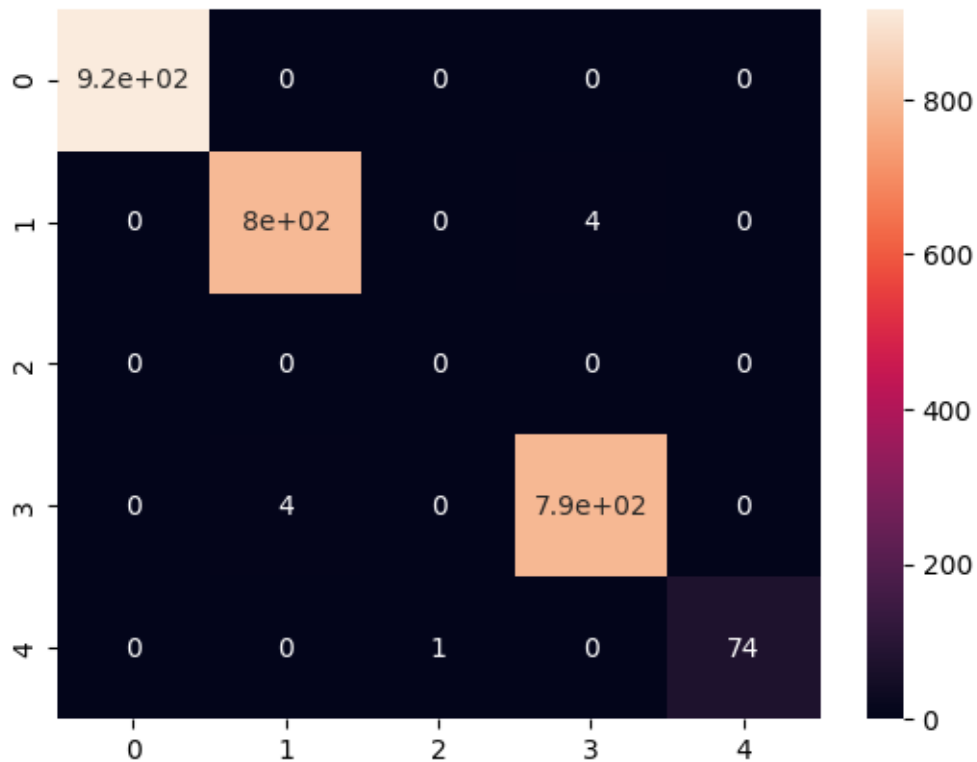
4.3 Kết quả cho tỉ lệ 80/20

	precision	recall	f1-score	support
not_recom	1.00	1.00	1.00	917
priority	1.00	1.00	1.00	801
recommend	0.00	0.00	0.00	0
spec_prior	0.99	0.99	0.99	798
very_recom	1.00	0.99	0.99	75
accuracy			1.00	2591
macro avg	0.80	0.80	0.80	2591
weighted avg	1.00	1.00	1.00	2591

Hình 4.11 Classifier report cho tỉ lệ 80/20

Với classifier report phía trên ta đưa ra được nhận xét rằng:

- Với lớp not_recom và priority, mô hình dự đoán đúng hoàn toàn với hệ số f1-score là 1.
- Lớp spec_prior và very_recom cũng được dự đoán với độ chính xác gần như tuyệt đối với hệ số f1-score cùng bằng 0.99
- Lớp recommend có hệ số f1-score bằng 0 vì ta không có đủ dữ liệu để train và test cho lớp này (tổng số 2 trên 12959)



Hình 4.12 Confusion matrix cho tỉ lệ 80/20

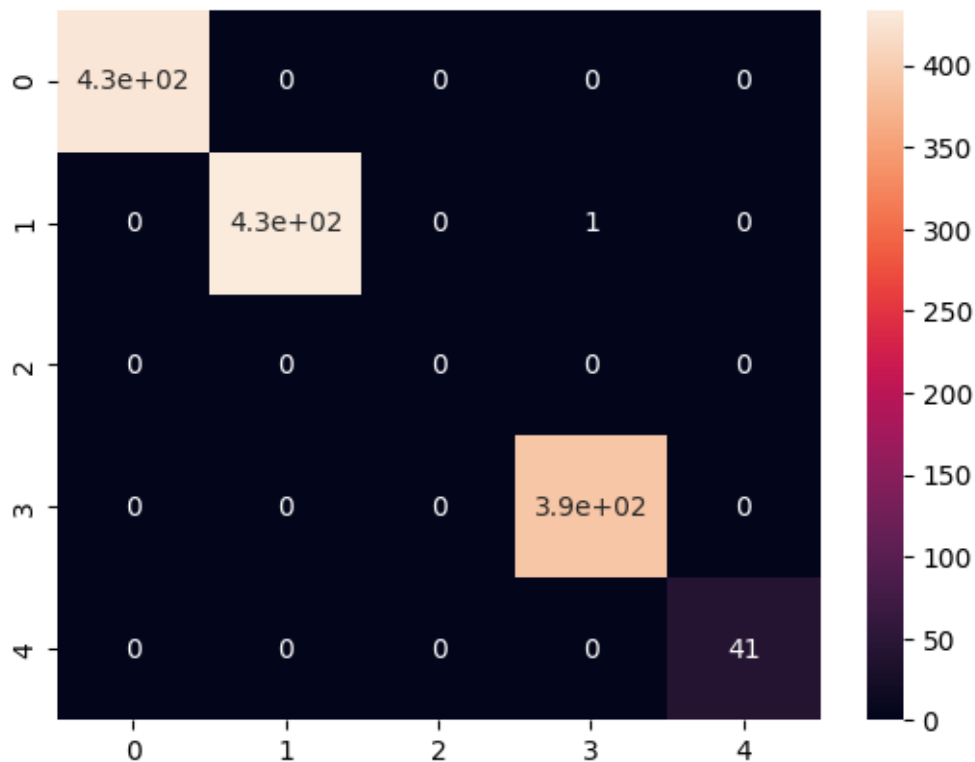
Theo như confusion matrix phía trên ta thấy có một số trường hợp được gán nhãn nhầm giữa nhãn 3 và 1. Chỉ có một trường hợp gán nhãn nhầm từ nhãn 2 qua nhãn 4 còn lại các nhãn đều được gán đúng.

4.4 Kết quả cho tỉ lệ 90/10

	precision	recall	f1-score	support
not_recom	1.00	1.00	1.00	428
priority	1.00	1.00	1.00	435
spec_prior	1.00	1.00	1.00	391
very_recom	1.00	1.00	1.00	41
accuracy			1.00	1295
macro avg	1.00	1.00	1.00	1295
weighted avg	1.00	1.00	1.00	1295

Hình 4.13 Classifier report cho tỉ lệ 90/10

Với classifier report phía trên ta đưa ra được nhận xét mô hình hoạt động rất tốt thông qua chỉ số f1-score của các lớp đều bằng 1. Điều này có thể sẽ khác với các lần chạy khác khi chúng ta cập nhật lại bộ dữ liệu cho các tập train và test



Hình 4.14 Confusion matrix cho tỉ lệ 90/10

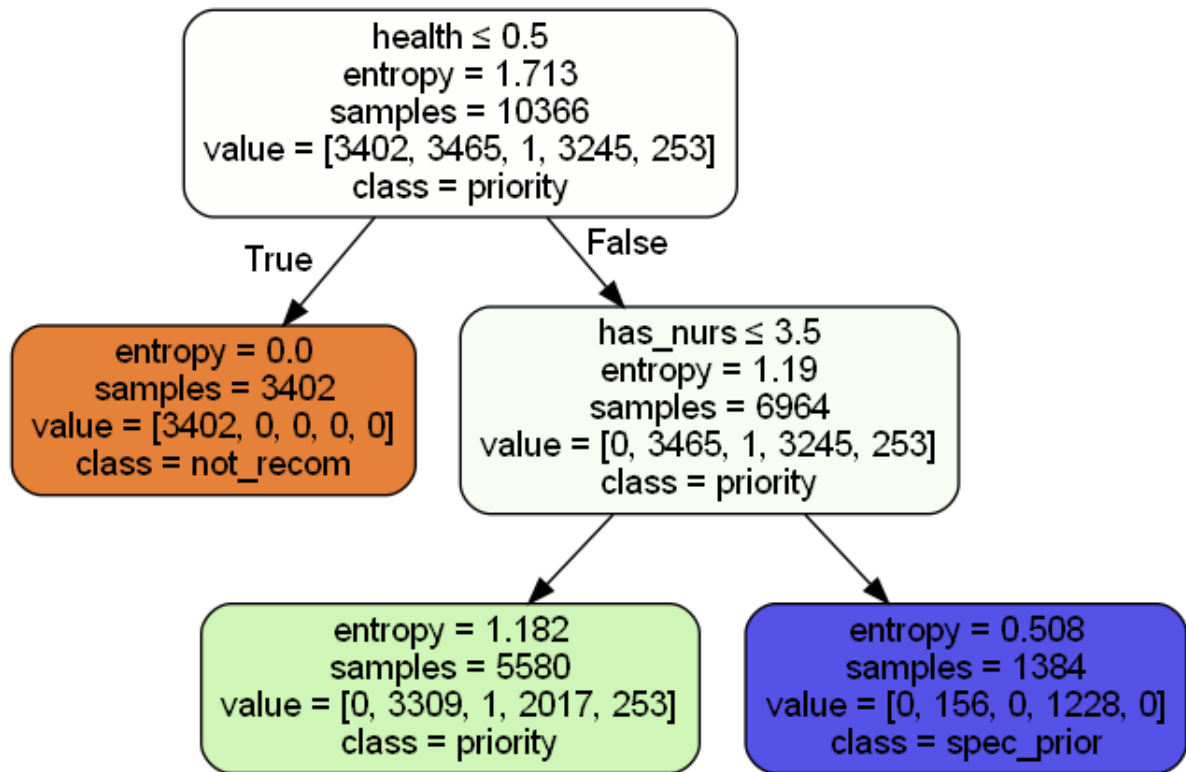
Theo như confusion matrix phía trên ta thấy chỉ có một trường hợp được gán nhãn nhầm từ nhãn 3 qua nhãn 1.

4.5 Nhận xét tổng quan

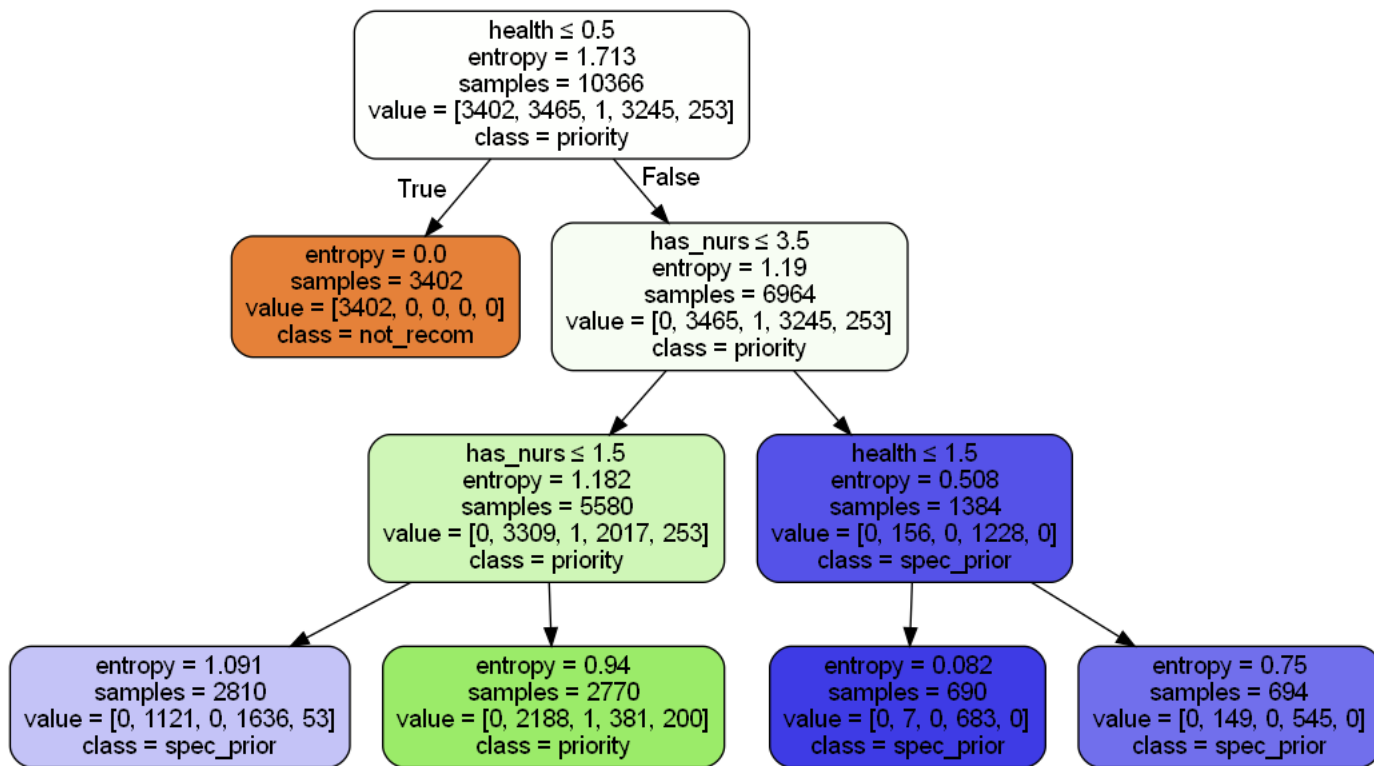
Cùng với độ mở rộng của tập train, mô hình phát triển càng ngày càng tốt lên với nhiều dữ liệu hơn từ tập train cho mô hình. Mô hình xuất phát với tập train tỉ lệ 40/60 đưa ra kết quả tệ nhất và kết quả này được cải thiện dần dần qua các tỉ lệ khác của tập train như 60/40, 80/20 và đạt một kết quả rất tốt ở tỉ lệ dữ liệu 90/10.

5 Độ sâu và độ chính xác của mô hình

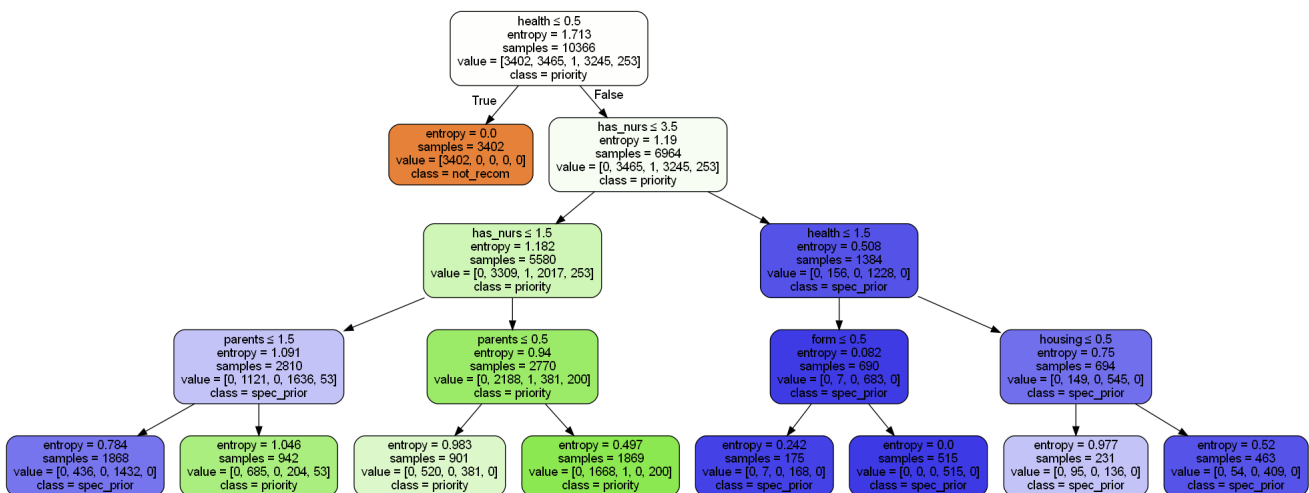
5.1 Trực quan hóa đồ thị



Hình 5.15 Mô hình cây với max_depth = 2



Hình 5.16 Mô hình cây với max_depth = 3



Hình 5.17 Mô hình cây với max_depth = 4

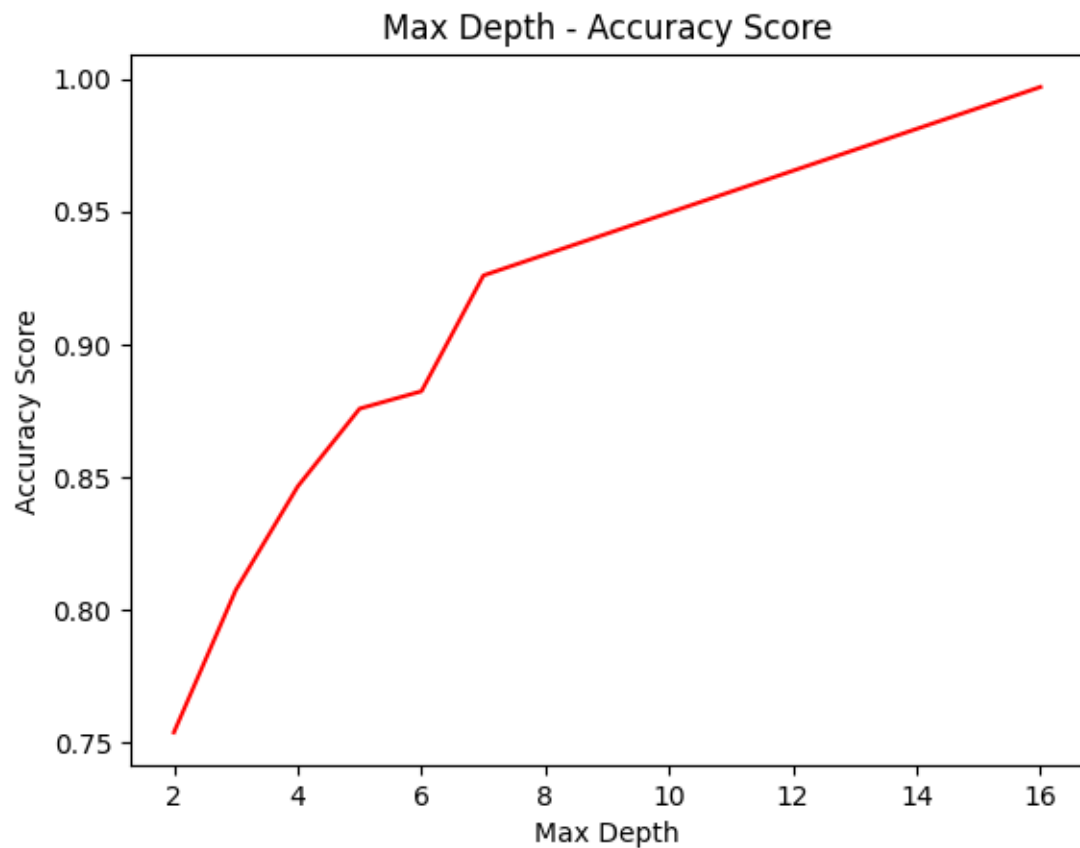
Với các mô hình có max_depth bằng 5, 6, 7 và None ta xem chi tiết trực quan tại đường dẫn `./Source/depth/graph_max_depth_5.png`, `./Source/depth/graph_max_depth_6.png`, `./Source/depth/graph_max_depth_7.png`, `./Source/depth/graph_max_depth_16.png`, hoặc trong file mã nguồn `21127021.ipynb`

5.2 Bảng thống kê

Bảng 5.2 Bảng thống kê ảnh hưởng của độ sâu đến độ sai lệch

max_depth	2	3	4	5	6	7	None
Accuracy	0.75376	0.80741	0.84639	0.87572	0.88228	0.9259	0.99691

5.3 Đồ thị trực quan hóa



Hình 5.18 Đồ thị ảnh hưởng giữa độ sâu và độ sai sót

5.4 Nhận xét

Từ những dữ kiện phía trên ta có thể đưa ra nhận xét rằng độ sâu của cây ảnh hưởng rất lớn đến độ chính xác của mô hình. Với cây khi có độ sâu thấp, mô hình trả về kết quả tệ hay nói cách khác mô hình đang bị vướng vào trường hợp *underfitting*. Khi ta tăng dần độ sâu của mô hình, kết quả accuracy score trả về cũng theo đó tăng lên. Ban đầu accuracy score tăng rất nhanh theo độ sâu (độ sâu 2 đến 5) nhưng sau đó tốc độ tăng của accuracy score đã giảm dần nhưng vẫn giữ ở mức tăng và đạt đỉnh khi ta để độ sâu là **None**. Lúc này

mô hình trả ta về một đồ thị với độ sâu bằng 16 và giá trị accuracy score là 0.99691, đây là một giá trị accuracy score rất tốt (tiệm cận với 1).