**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



**BÁO CÁO BÀI TẬP LỚN**

**MÔN: CHUYÊN ĐỀ HỆ THỐNG THÔNG TIN**

**Đề tài: *Sử dụng một cây quyết định cho data set marscrater\_pds.csv (Mars Craters codebook) trong khóa học***

|  |  |
| --- | --- |
| **Giảng viên**  **Nhóm lớp**  **Nhóm báo cáo** | **: Nguyễn Văn Tới**  **: 02**  **: Nhóm 08** |

**Các thành viên trong nhóm:**

|  |  |
| --- | --- |
| 1. **Nguyễn Nam Dương** | * **B16DCCN104** |
| 1. **Trần Thị Hải** | * **B16DCCN416** |
| 1. **Nguyễn Thu Huyền** | * **B16DCCN184** |
| 1. **Nguyễn Thị Xuân** | * **B16DCCN413** |

**HÀ NỘI – NĂM 2020**

# **MỤC LỤC**

[**MỤC LỤC** 2](#_heading=h.gjdgxs)

[**1.Phân công công việc, nhận xét, cho điểm của nhóm trưởng đối với các thành viên** 3](#_heading=h.1fob9te)

[**2. Đặt vấn đề** 4](#_heading=h.3znysh7)

[**3.Giới thiệu những công trình nghiên cứu đã có:** 5](#_heading=h.2et92p0)

[**4. Phương pháp cây quyết định** 6](#_heading=h.tyjcwt)

[**5. Phương pháp đã triển khai** 9](#_heading=h.3dy6vkm)

[**6. Mars Craters Study** 10](#_heading=h.1t3h5sf)

[**7.Chi tiết việc thí nghiệm** 11](#_heading=h.4d34og8)

[**8.Kết quả thu được** 17](#_heading=h.2s8eyo1)

[**9.Định hướng phát triển tiếp theo** 17](#_heading=h.17dp8vu)

[**10.Trình bày chi tiết việc triển khai, cài đặt:** 18](#_heading=h.3rdcrjn)

[**11.Những kết luận rút ra, những thu hoạch của các thành viên trong nhóm (kiến thức, kỹ năng, ...)** 28](#_heading=h.26in1rg)

* + - 1. **Phân công công việc, nhận xét, cho điểm của nhóm trưởng đối với các thành viên**

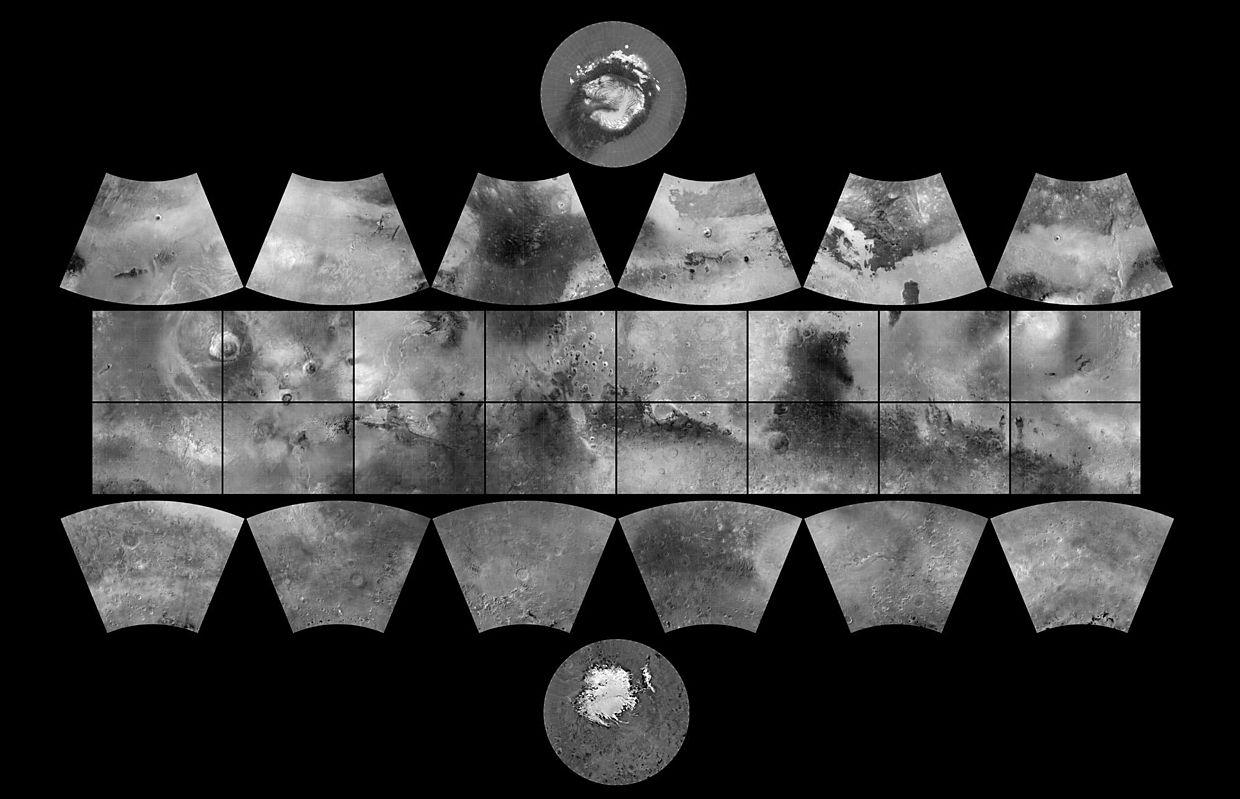
|  |  |  |
| --- | --- | --- |
| Tên Thành Viên | Đã làm | Đánh giá(tốt, chưa tốt) |
| Nguyễn Nam Dương | Tìm hiểu 4 loại phương pháp học máy trong học máy. | - Chưa tìm hiểu chi tiết về cả 4 phương pháp  - Mới tìm hiểu được chi tiết về cây quyết định |
| Nguyễn Thu Huyền | Tìm hiểu các course book trong khóa học | * Đã phần đã hiểu về bộ dữ liệu. * Nhiệt tình * Dịch tài liệu tiếng anh vẫn hơi trừu tượng |
| Nguyễn Thị Xuân | Tìm hiểu hai phương pháp đánh giá bias và trade off , tìm hiểu về cây quyết định | - Đã hiểu về cây quyết định  - Nhiệt tình  - Chủ động trong công việc |
| Trần Thị Hải | Tìm hiểu các khái niệm cơ bản của học máy | * Đã hiểu về khái niệm cơ bản của học máy * Nhiệt tình * Cần đóng góp ý kiến chủ động hơn |
| Dương Công Thành |  |  |

|  |  |  |
| --- | --- | --- |
| Tên Thành Viên | Thu hoạch | Kết luận rút ra |
| Nguyễn Nam Dương | - Hiểu về hoạt động của cây quyết định, đưa ra được câu hỏi tốt nhất | - Khá tốt: 7 điểm |
| Nguyễn Thu Huyền | - Hiểu về các thuộc tính và mô tả được các thuộc tính. Đưa ra được các ví dụ | - Khá tốt: 7 điểm |
| Nguyễn Thị Xuân | - Hiểu về khái niệm, mô hình bias  - Đưa ra được ví dụ trực quan của bias và variance và mối quan hệ của chúng | - Khá tốt: 7 điểm |
| Trần Thị Hải | - Hiểu hơn về học máy và quy trình xây dựng của mechine learning  - Cách phân loại và áp dụng | - Khá tốt: 7 điểm |
| Dương Công Thành |  |  |

# **2. Đặt vấn đề**

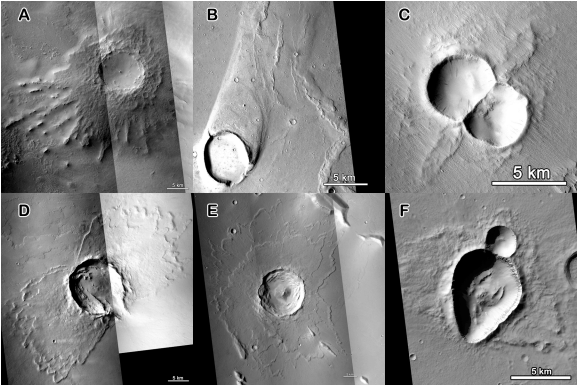
Bắt nguồn từ niềm cảm hứng muốn tìm hiểu về sao Hỏa, giống như trong các bộ phim viễn tưởng đã xem hồi còn nhỏ nên nhóm đã chọn một bộ dữ liệu liên quan đến hành tinh sao Hỏa. Bộ dữ liệu này có thể cho biết các vấn đề, sự kiện lịch sử đã xảy ra trên sao Hỏa, và điều đó đã tác động đến địa chất, khí hậu trên sao Hỏa như thế nào?

Bề mặt sao Hỏa đã được cơ quan Khảo sát địa chất Hoa Kỳ chia thành 30 hình tứ giác, và đường viền các hình tứ giác này nằm dọc theo các đường vĩ độ và kinh độ và ta sẽ có bản đồ thể hiện với các hình chữ nhật. Các tứ giác này được đặt tên theo đặc điểm về vị trí và được đặt tên với tiền tố là MC viết tắt của Mars Chart. Sau đây là một sơ đồ tưởng tượng của hành tinh sao Hỏa được chia thành 30 hình tứ giác được liên kết với nhau.



Hình ảnh được lấy từ Wikipedia

Ở mỗi vùng này trên sao Hỏa có rất nhiều miệng hố với đủ các hình dạng khác nhau các loại hố này được đặt tên theo các loài hoa, hay các con côn trùng mà hình dạng trông giống với loài hoa hay côn trùng đó. Sau đây là một số hình:



Nguồn ảnh từ mô tả bộ dữ liệu Mars Crater Study

Ở đây ta có thể thấy các hình dạng khác nhau của các miệng hố trên sao Hỏa, giả sử như hình C là trông giống hình con ong, hình F trong giống một còn bướm và chúng sẽ được đặt tên tương ứng là BumbleBee và ButterFly. Radial là một loại hình thái hay một đặc trưng thêm cho hình thái của miệng hố trên sao Hỏa.

Hình thái của miệng hố trên sao Hỏa có rất nhiều hình thái khác nhau nhưng ta chỉ phân lớp là Radial hay không phải Radial.

Vậy vấn đề cần giải quyết là:

* Dựa vào ví trí của các miệng hố này thuộc vào vị trí nào trong 30 vị trí trên sao Hỏa
* Dựa vào độ sâu của từng miệng hố
* Dựa vào số lớp, bán kính miệng hố trên sao Hỏa

Để xác định miệng hố trên sao Hỏa đó thuộc về phân lớp hình dạng là Radial hay không phải Radial.

* + - 1. **Giới thiệu những công trình nghiên cứu đã có:**

Công trình nghiên cứu này nằm trong khóa học Machine learning for data analysís được công bố bởi Coursera.

Mục đích nghiên cứu: Tìm hiểu xem các đặc điểm về vật lý như hình dạng độ sâu có liên quan đến hình thái của miệng núi lửa hay không?

Các trường thuộc tính sử dụng là:

* Quadrangle
* DIAM\_CIRCLE\_IMAGE
* DEPTH\_RIMFLOOR\_TOPOG
* NUMBER\_LAYERS

Để phân lớp thuộc tính trường MORPHOLOGY\_EJECTA\_1 là Radial hay không ở trong bài toán coi như là Yes và No

Tham số lựa chọn thuộc tính phân chia là Entropy để phát triển cây và để tỉa lá thì sử dụng cost-complexity.

Phương pháp: Sử dụng cây quyết định, thước đo phân chia Entropy, cây quyết định có hai nhánh, độ sâu tối đa là 10, số nhánh khi chưa tỉa là 505 sau khi tỉa là 171.

Kết quả:

Sử dụng cross validation tìm ra 171 lá

Sử dụng Confusion matrix thể hiện mô hình có thể dự đoán các miệng hố trên sao hỏa là Radial với tỉ 98% ( tỉ lệ đoán lỗi là 0.0124). Dự đoán đúng miệng hố trên sao hỏa là Radial kém hiệu quả chỉ có 42% ( tỉ lệ đoán lỗi là 0.5792). AUC là 0.8985 và Gini là 0.0808 cho tập dữ liệu huấn luyện.

Trường dữ liệu phân lớp quan trọng nhất là DEPTH\_RIMFLOOR\_TOPOG, biến đầu tiên được phân chia với giá trị ngưỡng là 0.010

|  |  |  |  |
| --- | --- | --- | --- |
| Thực tế | Dự đoán | |  |
| No | Yes | Error rate |
| No | 352829 | 4445 | 0.0124 |
| Yes | 15679 | 11390 | 0.5792 |

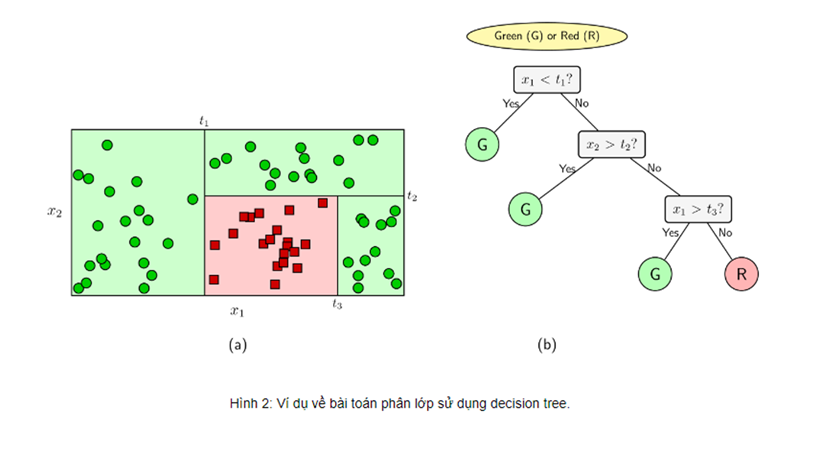
# **4. Phương pháp cây quyết định**

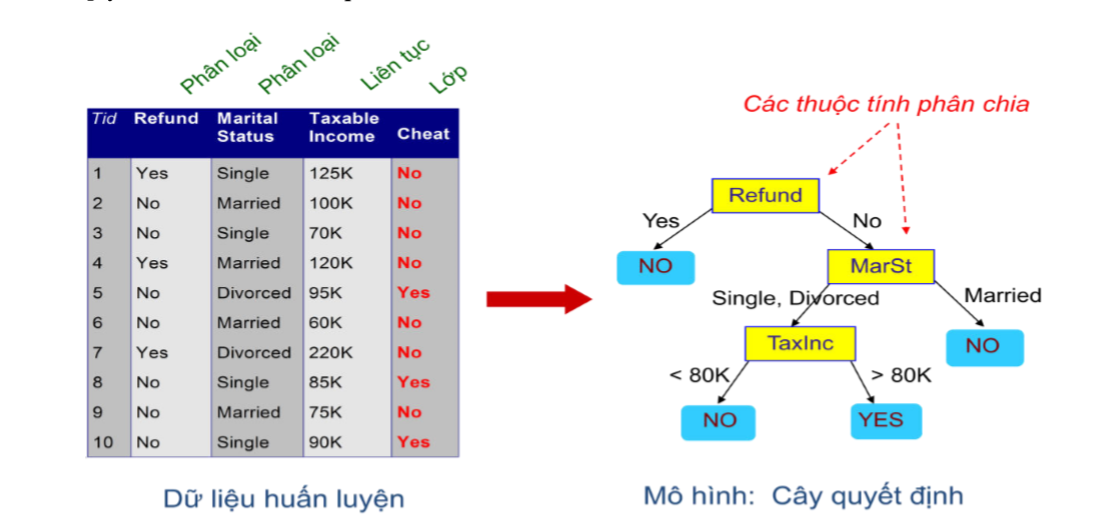
Cây quyết định là gì?

* Cây quyết định (Decision Tree) là một mô hình thuộc nhóm thuật toán học có giám sát (Supervised Learning).
* Thuật toán này có thể được sử dụng để giải quyết cả hai bài toán hồi quy (Regression) và phân loại (Classification).

Trong cây quyết định

* Các node thể hiện đầu ra được gọi là node lá (leaf node hoặc terminal node).
* Các node thể hiện câu hỏi là các non-leaf node.
* Non-leaf node trên cùng (câu hỏi đầu tiên) được gọi là node gốc (root node).
* Các non-leaf node thường có hai hoặc nhiều node con (child node). Các child node này có thể là một leaf node hoặc một non-leaf node khác.
* Các child node có cùng bố mẹ được gọi là sibling node.
* Nếu tất cả các non-leaf node chỉ có hai child node, ta nói rằng đó là một binary decision tree (cây quyết định nhị phân).
* Các câu hỏi trong binary decision tree đều có thể đưa được về dạng câu hỏi đúng hay sai.

1. Ví dụ bài toán phân lớp sử dụng cây quyết định



Các thuật toán tìm cây quyết định

Có nhiều nhóm thuật toán được áp dụng để xây dựng cây quyết định bao gồm

* Các thuật toán của Hunt: là một nhóm thuật toán ra đời sớm nhất
* Nhóm thuật toán CART
* Nhóm thuật toán ID3, C4.5
* Nhóm SLIQ , SPRINT

Thuật toán ID3

ID3 cho phép chúng ta xác định phân loại của một đối tượng bằng cách kiểm tra các giá trị của nó trên một số thuộc tính nào đó. Nhiệm vụ của giải thuật ID3 là học cây quyết định từ một tập training set

Input: Một tập hợp các mẫu. Mỗi mẫu bao gồm các thuộc tính mô tả một tình huống, hay một đối tượng nào đó, và một giá trị phân loại (thẻ) của nó.

Output: Cây quyết định có khả năng phân loại đúng đắn các mẫu trong training set, và hy vọng là phân loại đúng cho cả các ví dụ chưa gặp trong tương lai.

Giải thuật ID3 xây dựng cây quyết định được trình bày như sau:

(Xây dựng lần lượt các nút của cây bắt đầu từ gốc)

+ Khởi đầu: nút hiện thời là nút gốc chứa toàn bộ tập dữ liệu huấn luyện

+ Tại nút hiện thời n, lựa chọn thuộc tính

Chưa được sử dụng ở nút tổ tiên

Cho phép phân chia tập dữ liệu hiện thời thành các tập con 1 cách tốt nhất

Với mỗi giá trị thuộc tính được chọn thêm 1 nút con dưới

Chia các ví dụ ở nút hiện thời về các nút con theo giá trị thuộc tính được chọn

+ Lặp cho tới khi:

Tất cả các thuộc tính đã được sử dụng ở các nút phía trên

Tất cả ví dụ tại nút hiện thời có cùng nhãn phân loại

Nhãn của nút được lấy theo đa số nhãn của ví dụ tại nút hiện thời

Tiêu chuẩn chọn nút của ID3:

+ Tại mỗi nút n: Cần lựa chọn thuộc tính cho phép phân chia tập dữ liệu tốt nhất

+ Tiêu chuẩn:

Dữ liệu sau khi phân chia càng đồng nhất càng tốt

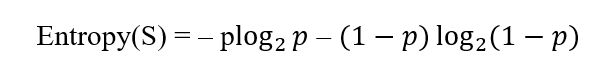
Đo bằng độ tăng thông tin (Information Gain –IG)

Chọn thuộc tính có độ tăng thông tin lớn nhất

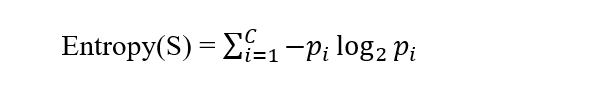
IG dựa trên Entropy của tập dữ liệu

Entropy

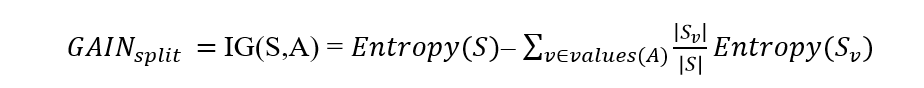
+ Trường hợp có 2 loại nhãn: đúng, sai



+ Trường hợp tổng quát: có C loại nhãn

 Độ tăng thông tin (Information Gain)

+ Thông tin (Information) là cái hiểu biết về đối tượng, làm rõ đối tượng



Trong đó: values(A) là tập các giá trị của A

S\_v là tập con của S bao gồm các mẫu có giá trị của A = v

|S| là số phần tử của S

Ưu điểm: Dễ cài đặt

Nhược điểm:

- Độ chính xác không cao

- Số nhánh trong cây quyết định tăng thì GAIN cũng tăng

- Không thực hiện được với dữ liệu liên tục

# **5. Phương pháp đã triển khai**

Sử dụng cây quyết định có kĩ thuật sau:

* Thước đo phân chia: Entropy
* Hỗ trợ cho cả dữ liệu rời rạc và liên tục
* Sử dụng tiền tỉa lá vào các trường hợp các bản ghi thuộc cùng một lớp, giới hạn độ sau là 10, hoặc không còn bản ghi nào.
* Nếu dữ liệu là rời rạc thì chỉ được sử dụng để phân chia một lần nghĩa là các nút con sẽ không được sử dụng lại thuộc tính rời rạc mà cha nó đã sử dụng.

Sử dụng thước đo độ chính xác:

* Sử dụng confusion matrix đã chuẩn hóa về 0 và 1
* Tính recall và precision
* Tính độ bỏ sót (FNR) và độ đoán chính xác (TPR)

Phương pháp chọn mô hình tốt nhất: Sử dụng k fold cross validation tức chia tập dữ liệu thành k phần với k là số do ta chọn. k-1 phần được dùng để huấn luyện, còn 1 phần để đo độ chính xác. Sẽ chọn k tất cả từ 2 đến 10 và tìm ra mô hình có độ chính xác phù hợp với bài toán ví dụ trong tập dữ liệu sao hỏa có rất nhiều bản ghi không phải Radial việc dự đoán đúng lớp No sẽ không được ưu tiên bằng việc đoán là Yes.

# **6. Mars Craters Study**

Địa hình hay bề mặt của sao Hỏa có rất nhiều các miệng hố được tạo ra từ 4.2 đến 3,8 tỷ năm trước. Trong thời kỳ đó có nhiều các thiên thạch va đập với sao Hỏa (tức là tác động của các tiểu hành tinh, hành tinh nguyên sinh, và sao chổi). Nhưng từ đó đến này, bề mặt của Sao Hỏa không thay đổi nhiều. Các bề mặt trên các vật thể không có không khí như Sao Hỏa nên các bề mặt và các miệng hố của sao Hỏa không thay đổi nhiều qua thời gian. Các miệng hố xuất hiện trên toàn bộ bề mặt Sao Hỏa và chúng đóng vai trò quan trọng giúp ta hiểu tính chất vỏ của Sao Hỏa cũng như tuổi bề mặt và các sự kiện đã xảy ra trong quá khứ. Chúng cho phép suy đoán về khí hậu cổ đại và lịch sử, và thêm một điểm dữ liệu quan trọng cho sự hiểu biết về vật lý va chạm giữa các tiểu hành tinh với sao hỏa như thế nào. Nghiên cứu này, được tạo bởi Stuart Robbins, trình bày một cơ sở dữ liệu toàn cầu mới cho Sao Hỏa chứa 378.540 miệng hố thống kê cho đường kính D=1 (km).

Các thuộc tính trong Mars Craters

• CRATER\_ID - ID miệng núi lửa cho vụ kiện nội bộ, dựa trên khu vực của hành tinh

• LATITUDE\_CIRCLE\_IMAGE - vĩ độ từ tâm xuất phát của phi tuyến tính vòng tròn hình vuông nhỏ nhất vừa với các đỉnh được chọn để nhận dạng thủ công vành miệng núi lửa

(đơn vị là độ thập phân Bắc)

• LONGITUDE\_CIRCLE\_IMAGE - kinh độ từ tâm xuất phát của phi tuyến tính vòng tròn hình vuông nhỏ nhất vừa với các đỉnh được chọn để nhận dạng thủ công vành miệng núi lửa

(đơn vị là độ thập phân Đông)

• DIAM\_CIRCLE\_IMAGE - đường kính từ một vòng tròn bình phương nhỏ nhất phi tuyến tính phù hợp với các đỉnh được chọn để xác định thủ công vành miệng núi lửa (đơn vị là km)

• DEPTH\_RIMFLOOR\_TOPOG - độ cao trung bình xác định N điểm dọc theo (hoặc bên trong) vành miệng núi lửa (đơn vị là km)

o Depth Rim - Điểm được chọn là mức cao địa hình tương đối bên dưới giả định chúng là ít bị xói mòn nhất vì vậy hầu hết các điểm ban đầu vành miệng núi lửa

o Depth Floor- Điểm được chọn là độ cao thấp nhất không bao gồm các miệng hố nhúng có thể nhìn thấy

• MORPHITALY\_EJECTA\_1 - hình thái cuội núi lửa được phân loại. Ví dụ dưới đây.

o Nếu có nhiều giá trị, được phân tách bằng một giá trị / dòng, thì thứ tự là cuội núi lửa bên trong nhất thông qua bên ngoài nhất, hoặc trên cùng nhất dưới cùng nhất

• MORPHITALY\_EJECTA\_2 - hình thái của chính (các) lớp đó.

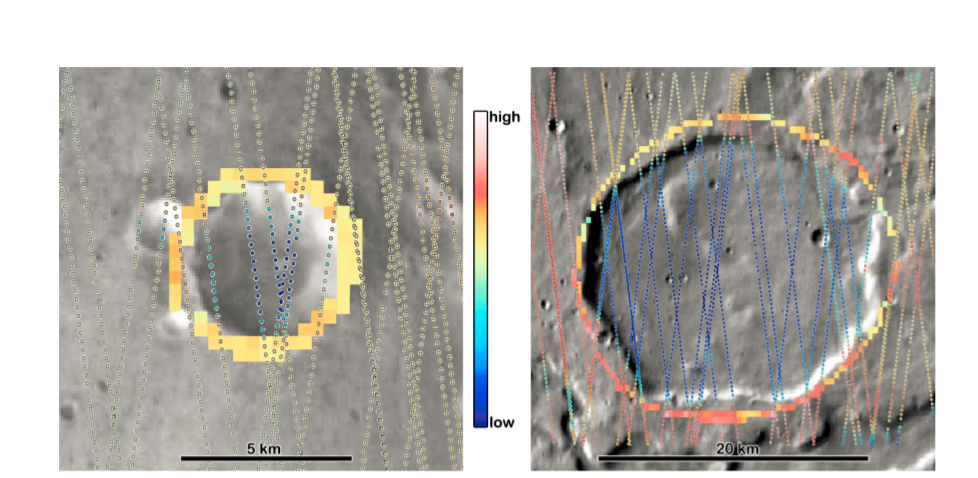
Hệ thống phân loại này là duy nhất cho công việc này. Ví dụ dưới đây.

• MORPHITALY\_EJECTA\_3 - kết cấu tổng thể và / pr hình dạng của một số

lớp (cuội núi lửa) nói chung là duy nhất và xứng đáng có hình thái riêng biệt

phân loại. Ví dụ dưới đây.

• NUMBER\_LAYERS - số lớp kết dính tối đa trong bất kỳ phương vị nào hướng có thể được xác định một cách đáng tin cậy



1. **Chi tiết việc thí nghiệm**

Cách đánh giá vì bộ dữ liêu số lượng lớp Rd là 27626 chỉ chiểm 7%, và Not Rd là rất nhiều, cùng với việc để tránh overfitting nên thuật toán cây quyết định cần giới hạn độ sâu là 10, tỉ lệ lớp nào lớn hơn sẽ phải trả về là lớp đó, nên việc tỉ lệ đoán đúng là Not Rd thường sẽ rất cao, tỉ lệ Rd sẽ thấp hơn rất nhiều. Việc đúng Not Rd vì có nhiều dữ liệu hơn rất nhiều so với Rd sẽ không đánh giá tốt được bằng tỉ lệ đúng Rd. Vì vậy ta chỉ đánh giá mô hình dựa trên tỉ lệ liên quan đến dự đoán đúng Rd hay Yes mà thôi cụ thể ở đây là chỉ số FNR là tỉ lệ bỏ sót và tỉ lệ FPR là tỉ lệ dự đoán sai là hai thông số sẽ được sử dụng để đánh giá dọc suốt quá trình thử nghiệm.

**Thử bỏ thuộc tính DIAM\_CIRCLE\_IMAGE ( thời gian chạy mỗi mô hình thường là 8 phút)**

**Chia 384343 bản ghi thành 3 phần để được 3 mô hình, mỗi mô hình lấy 2 phần làm dữ liệu huấn luyện, 1 phần dùng để validation( kiểm tra).**

Kết quả (% ) thu được như sau:

**Sử dung 1 phần đầu test, hai phần sau huấn luyện**

|  |  |  |
| --- | --- | --- |
| Thực tế | Dự đoán | |
| No | Yes |
| No | 0.9883845008114176 | 0.011615499188582352 |
| Yes | 0.6756005883314267 | 0.3243994116685733 |

Recall của Yes: 0.3243994116685733

Precision của Yes: 0.6551155115511551

Accuracy: 94.60998074621429

Ta thấy với trường hợp này tỉ lệ FNR (tỉ lệ bỏ sót Radial) rất cao khoảng 67%, tỉ lệ recall rất thấp, tỉ lệ Precision khá cao.

**Sử dung phần giữa test, hai phần sau huấn luyện**

|  |  |  |
| --- | --- | --- |
| Thực tế | Dự đoán | |
| No | Yes |
| No | 0.9767886618475724 | 0.023211338152427572 |
| Yes | 0.5462659141619635 | 0.4537340858380365 |

Recall của Yes: 0.4537340858380365

Precision của Yes: 0.6290029112081513

Accuracy: 92.90003642608107

Ta thấy với trường hợp này tỉ lệ FNR (tỉ lệ bỏ sót Radial) khoảng 54% vẫn khá là cao nhưng đã thấp hơn so với trường hợp trên, tỉ lệ recall đã cao hơn một chút.

**Sử dung phần cuối test, hai phần sau huấn luyện**

|  |  |  |
| --- | --- | --- |
| Thực tế | Dự đoán | |
| No | Yes |
| No | 0.976474278544542 | 0.023525721455457966 |
| Yes | 0.5021257880076235 | 0.49787421199237647 |

Recall của Yes: 0.49787421199237647

Precision của Yes: 0.6179039301310044

Accuracy: 94.24988291616798

Ta thấy với trường hợp này tỉ lệ FNR (tỉ lệ bỏ sót Radial) khoảng 50% vẫn khá là cao nhưng đã thấp hơn so với cả 2 trường hợp trên, tỉ lệ recall cũng cao hơn cả hai trường hợp trên, tỉ lệ chính xác cũng cao hơn.

Nên ta sẽ chọn mô hình cuối này đối với k trong cross validation là 3.

**Chia 384343 bản ghi thành 5 phần để được 5 mô hình, mỗi mô hình lấy 4 phần làm dữ liệu huấn luyện, 1 phần dùng để validation( kiểm tra).**

**Sử dung phần 1 test, phần còn lại huấn luyện**

|  |  |  |
| --- | --- | --- |
| Thực tế | Dự đoán | |
| No | Yes |
| No | 0.9890403709841384 | 0.01095962901586152 |
| Yes | 0.694801344139158 | 0.30519865586084205 |

Recall của Yes: 0.30519865586084205

Precision của Yes: 0.6623766623766624

Accuracy: 94.40339282926575

Tỉ lệ FNR là 0.69 -> đây không phải mô hình tốt

**Sử dụng phần 2 test, phần còn lại huấn luyện**

|  |  |  |
| --- | --- | --- |
| Thực tế | Dự đoán | |
| No | Yes |
| No | 0.9987715161185557 | 0.0012284838814442451 |
| Yes | 0.9918994875185981 | 0.008100512481401884 |

Recall của Yes: 0.008100512481401884

Precision của Yes: 0.3602941176470588

Accuracy: 92.08123016079513

Tỉ lệ FNR là 0.99 -> đây không phải mô hình tốt

**Sử dụng phần 3 test, phần còn lại huấn luyện**

|  |  |  |
| --- | --- | --- |
| Thực tế | Dự đoán | |
| No | Yes |
| No | 0.9913122712205238 | 0.008687728779476222 |
| Yes | 0.6904109589041096 | 0.3095890410958904 |

Recall của Yes: 0.3095890410958904

Precision của Yes: 0.5507717303005687

Accuracy: 82.97080709788209

Tỉ lệ FNR là 0.69 -> đây không phải mô hình tốt

**Sử dụng phần 4 test, phần còn lại huấn luyện**

|  |  |  |
| --- | --- | --- |
| Thực tế | Dự đoán | |
| No | Yes |
| No | 0.980 | 0.019 |
| Yes | 0.5957 | 0.4042 |

Recall của Yes: 0.4042912654164555

Precision của Yes: 0.6294055760126249

Accuracy: 93.57990321069886

**Sử dụng phần 5 test, phần còn lại huấn luyện**

|  |  |  |
| --- | --- | --- |
| Thực tế | Dự đoán | |
| No | Yes |
| No | 0.9732 | 0.0267 |
| Yes | 0.6923 | 0.3076 |

Recall của Yes: 0.30763555883437843

Precision của Yes: 0.4664429530201342

Accuracy: 92.63412603424051

*Nhận xét:*  Ta thấy đa phần các mô hình ứng với k=5 đều cho tỉ FNR rất cao, recall và precision đều thấp hơn -> các mô hình đều không tốt hơn so với k=3

**Chia 384343 bản ghi thành 7 phần để được 7 mô hình, mỗi mô hình lấy 6 phần làm dữ liệu huấn luyện, 1 phần dùng để validation( kiểm tra).**

**Sử dung phần 1 test, phần còn lại huấn luyện**

|  |  |  |
| --- | --- | --- |
| Thực tế | Dự đoán | |
| No | Yes |
| No | 0.9896151736029469 | 0.01038482639705306 |
| Yes | 0.7777777777777778 | 0.2222222222222222 |

Recall của Yes: 0.2222

Precision của Yes: 0.61870

Accuracy: 93.5526

**Sử dung phần 2 test, phần còn lại huấn luyện**

|  |  |  |
| --- | --- | --- |
| Thực tế | Dự đoán | |
| No | Yes |
| No | 0.9974 | 0. 0025524 |
| Yes | 0.9896 | 0. 0103170 |

Recall của Yes: 0.0103170

Precision của Yes: 0.23976

Accuracy: 92.600080

**Sử dung phần 3 test, phần còn lại huấn luyện**

|  |  |  |
| --- | --- | --- |
| Thực tế | Dự đoán | |
| No | Yes |
| No | 0.999 | 0. 007 |
| Yes | 0.999 | 0.000 |

Recall của Yes: 0.00053

Precision của Yes: 0.3333

Accuracy: 93.1719

**Sử dung phần 4 test, phần còn lại huấn luyện**

|  |  |  |
| --- | --- | --- |
| Thực tế | Dự đoán | |
| No | Yes |
| No | 1.0 | 0.0 |
| Yes | 1.0 | 0.0 |

Recall của Yes: 0.0

Precision của Yes: 0

Accuracy: 86.19094452336721

**Sử dung phần 5 test, phần còn lại huấn luyện**

|  |  |  |
| --- | --- | --- |
| Thực tế | Dự đoán | |
| No | Yes |
| No | 0.970 | 0.0296454754777804 |
| Yes | 0.382 | 0.6416 |

Recall của Yes: 0.6175

Precision của Yes: 0.6416

Accuracy: 94.2

**Sử dung phần 6 test, phần còn lại huấn luyện**

|  |  |  |
| --- | --- | --- |
| Thực tế | Dự đoán | |
| No | Yes |
| No | 0.98 | 0.019 |
| Yes | 0.728448275862069 | 0.271 |

Recall của Yes: 0.2715

Precision của Yes: 0.501

Accuracy :93.25392488981167

**Sử dung phần 7 test, phần còn lại huấn luyện**

|  |  |  |
| --- | --- | --- |
| Thực tế | Dự đoán | |
| No | Yes |
| No | 0.9695185792457676 | 0.030 |
| Yes | 0.6765 | 0.3234539089848308 |

Recall của Yes: 0.323

Precision của Yes: 0.473

Accuracy : 91.909

*Nhận xét*:

Ta thấy có một trường hớp đặc biệt là phần test số 4 chỉ dự đoán 1 bản ghi nguyên nhân là do dữ liệu huấn luyện khi xây dựng cây quyết định tại nút đầu tiên là dữ liệu rời rạc quadrangle thì các nút không có các giá trị của nào ứng với các giá trị ở cột Quadrangle giống.

Ta thấy là có phần test 5 có tỉ lệ FNR (tỉ lệ bỏ sót ) thấp nhất là 0.38 vả tỉ lệ đoán nhầm là 0.029 khá thấp và tỉ lệ recall, precison cũng cao nhất-> đây là mô hình tốt nhất với k=7

**Sử dụng tất cả các thuộc tính như phân tích ban đầu (thời gian huấn luyện 1h30 phút với k=3):**

**Chia 384343 bản ghi thành 3 phần để được 3 mô hình, mỗi mô hình lấy 2 phần làm dữ liệu huấn luyện, 1 phần dùng để validation( kiểm tra).**

Kết quả (% ) thu được như sau:

**Sử dung 1 phần đầu test, hai phần sau huấn luyện**

|  |  |  |
| --- | --- | --- |
| Thực tế | Dự đoán | |
| No | Yes |
| No | 0.98377 | 0.01622607 |
| Yes | 0.46014 | 0.53985507 |

Recall của Yes: 0.5398550724637681

Precision của Yes: 0.7180198598130841

Accuracy: 95.22144340197012

**Sử dung phần giữa test, hai phần sau huấn luyện**

|  |  |  |
| --- | --- | --- |
| Thực tế | Dự đoán | |
| No | Yes |
| No | 0.98386 | 0.01613832 |
| Yes | 0.33585 | .0.6641562 |

Recall của Yes: 0.66415

Precision của Yes: 0.75043

Accuracy: 92.9

**Sử dung phần cuối test, hai phần sau huấn luyện**

|  |  |  |
| --- | --- | --- |
| Thực tế | Dự đoán | |
| No | Yes |
| No | 0.9902 | 0.0097 |
| Yes | 0.62920 | 0.3707 |

Recall của Yes: 0.37079

Precision của Yes: 0.75070

Accuracy: 94.4

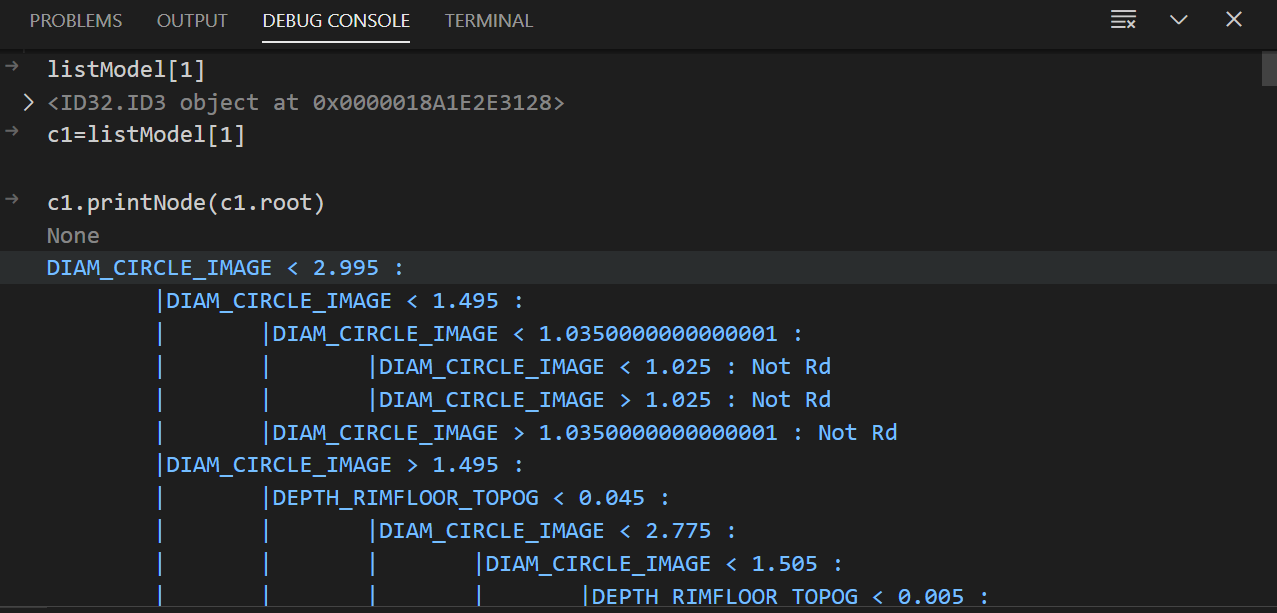
*Nhận xét*:

* Nhận xết ta thấy tỉ lệ FNR, FPR trung bình là thấp hơn so với khi bỏ thuộc tính DIAM\_CIRCLE\_IMAGE nên tốt hơn
* Ta thấy mô hình phần hai test có các độ đo FNR là thấp nhất và recall , precision là cao nhất nên đây là mô hình tốt nhất

*Nhận xét toàn bộ:*

* Khi so sánh k=3 có thuộc tính DIAM\_CIRCLE\_IMAGE tỉ lệ FNR chỉ là 33% còn không có thì 50%, tỉ lệ cả recall và precision đều cao hơn, nên ta rút ra là không chỉ thuộc tính độ sâu mà thuộc tính DIAM\_CIRCLE\_IMAGE cũng ảnh hưởng rất nhiều đến hình thái của miệng hố trên sao hỏa
* Xét các mô hình không có thuộc tính DIAM\_CIRCLE\_IMAGE, mô hình tốt nhất ở k=3 tốt hơn nhiều so với k=5 nhưng lại thấp hơn k=7, ta có thể thấy tỉ lệ dữ liệu huấn luyện không quyết định mô hình tốt hơn.

Hình ảnh cây tốt nhất mà nhóm tìm được:



1. **Kết quả thu được**

* Với thuật toán cây quyết định tuy về mặt lý thuyết là ít được dùng với dữ liệu quá lớn, nhưng thực tế vẫn có thể được sử dụng và khai phá dữ liệu khá hiệu quả và trực quan.
* Đã tìm được mô hình với độ chính xác là 93 % hầu như các mô hình đều có độ chính xác tương đương nên tập trung vào 2 chỉ số FNR và TPR, tìm ra mô hình có FNR xấp xỉ 33% thấp nhất tỉ lệ bỏ sót lớp Radial là ít nhất.
* Đã hiểu hơn về bộ dữ liệu sao Hỏa, biết được thuộc tính DIAM\_CIRCLE\_IMAGE cũng rất quan trọng trong việc xác định hình thái.
* Tìm được mô hình tốt hơn bài nghiên cứu đã có của Coursera với độ TPR cao hơn là 66-42=24% tức là k=3. Thuộc tính DIAM\_CIRCLE\_IMAGE là thuộc tính được phân nhánh đầu tiên nên đóng góp nhiều thông tin nhất.

1. **Định hướng phát triển tiếp theo**

* Tiếp tực phát triện thuật toán cây quyết định của nhóm thêm ví dụ sử dụng thêm các thước đo độ trong suốt khác như gini, gain ratito,… để so sánh và quan sát xem cấu trúc cây độ chính xác có thay đổi hay không.
* Tiếp tục phát triển thuật toán này từ ID3 thành C45 và C5.
* Tiếp tục sử dụng các thước đo lựa chọn mô hình khác như ROC Curve, AUC,…
* Sử dụng thuật toán này không chỉ trên bộ data set mar crater mà với nhiều bộ dữ liệu khác trong khóa học Machine Learning for data analys.
* Tiếp tục so sánh với các thuật toán phân lớp khác như bayes, neural network,...

Nguyễn Nam Dương: Muốn tiếp tục ứng dụng thuật toán cây quyết định này cho các bài toán về hệ thống thông tin quản lý cụ thể là phần mềm bán hàng, siêu thi hay các dữ liệu về bán hàng bằng việc tiếp tục đi tìm các dữ liệu thật để nghiên cứu. Từ việc khai phá dữ liệu bán hàng sẽ tạo thêm các chức năng gợi ý mặt hàng cho phần mềm bán hàng, dự báo lượng hàng kho phải nhập, nghiên cứu thêm thuật toán kmean để nhóm các mặt hàng hay bán cùng nhau,… muốn tìm hiểu xem chức năng này sẽ chạy trên một hệ thống phân tán như thế nào.

1. **Trình bày chi tiết việc triển khai, cài đặt:**

**Triển khai và cài đặt:**

Môi trường:

1. Phần mềm hỗ trợ cài đặt là visual studio code
2. Ngôn ngữ Python

Tiến hành cài đặt:

Bước 1:

Tải và cài đặt visual studio code.

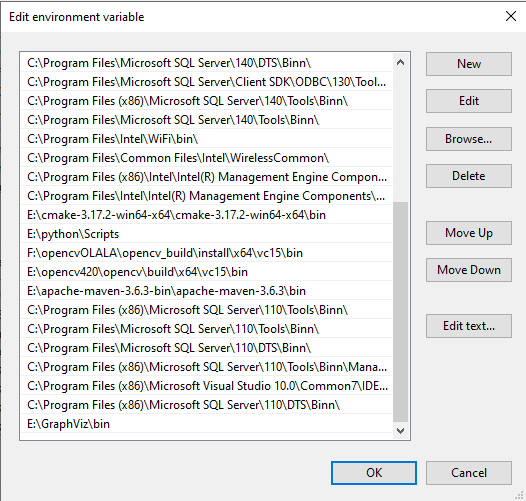


Tiến hành cài đặt extension python

Bước 2:

Download python và tiến hành cài đặt

Cài đặt biến môi trường cho python



Bước 3: Triển khai viết chương trình

Từ cửa sổ terminal gõ 3 dòng lệnh sau để tạo thư mục



Tạo file a.python và file Id3.python bằng click vào biểu tượng



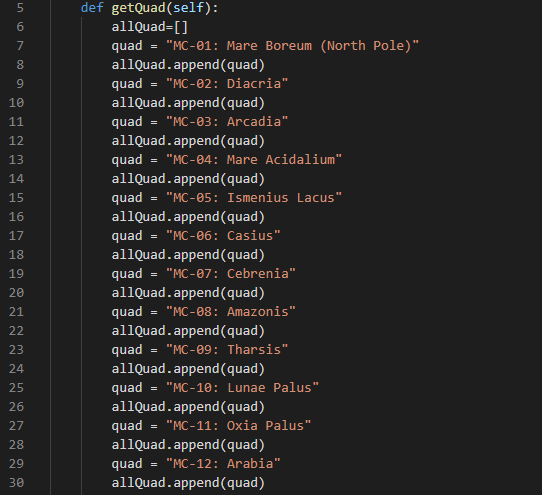
Đặt file mar.csv cùng vị trí với hai file trên

**Mô tả hệ thống:**

Các module chính của hệ thống:

1. Đọc dữ liệu đầu vào tạo dữ liệu cột dẫn xuất Quadrangle và cột phân lớp
2. Xây dựng cây quyết định cho cả dữ liệu liên tục và rời rạc
3. Sử dụng cross validation tìm ra mô hình tốt nhất.

**Module: Đọc dữ liệu đầu vào và tạo dữ liệu cột dẫn xuất Quadrangle và cột phân lớp**

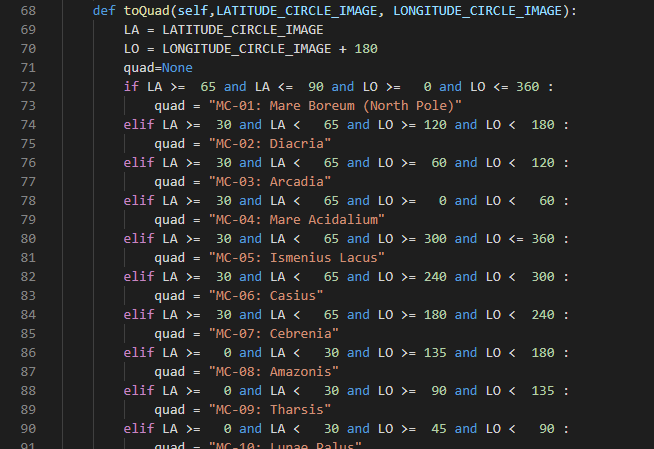


Hàm getQuad:

Đầu vào: không có

Đầu ra: Trả về danh sách các giá trị ở cột Quadrangle

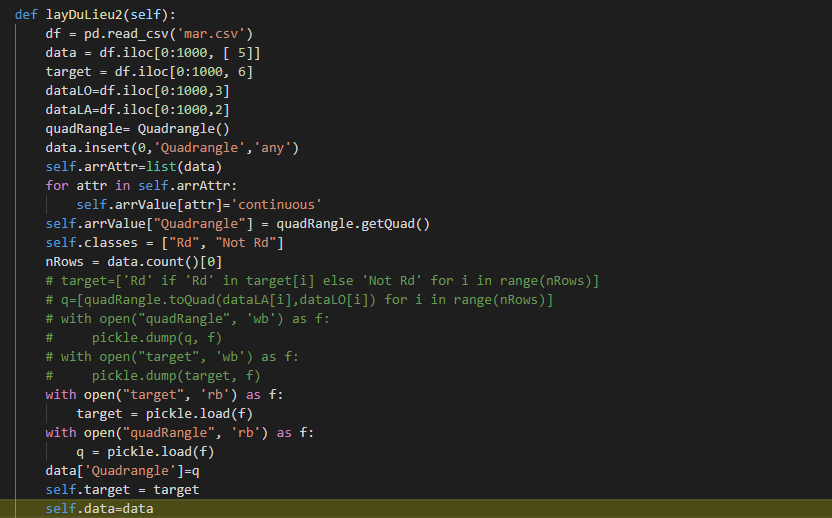
Mục đích: sử dụng để lưu các giá trị rời rạc của cột Quadrangle



Hàm toQuad:

Đầu vào: giá trị Kinh độ và vĩ độ ( tức 2 trường LATITUDE\_CIRCLE\_IMAGE và LONGTITUDE\_CIRCLE\_IMAGE)

Đầu ra: trả về 1 trong 30 vị trị ứng kinh độ và vĩ độ



Hàm layDuLieu2:

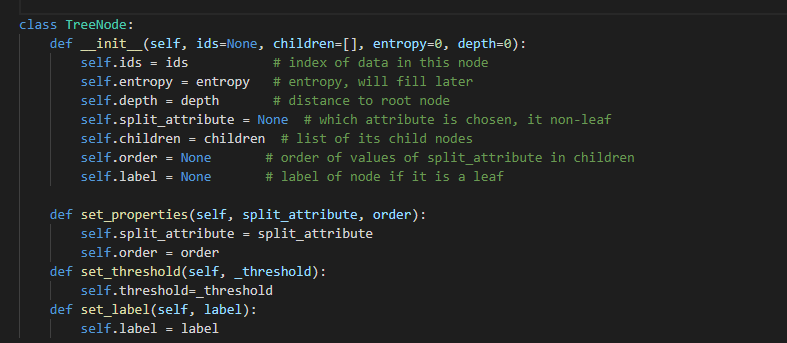
Đầu vào: dữ liệu chưa được được xử lý

Đầu ra: dữ liệu đã được xử lý là thêm cột dẫn xuất Quadrangle, và cột phân lớp có hai giá trị Rd và Not Rd, đánh dấu các cột là thuộc tính rời rạc hay là liên tục

Mục đích: Tạo dữ liệu để tạo mô hình bằng cây quyết định ứng với dữ liệu dó.

**Module: Xây dựng cây quyết định cho cả dữ liệu liên tục và rời rạc**

File ID32.py chứa class định nghĩa các thuộc tính của một node trong cây quyết định



TreeNode là tên nút trong cây quyết định

Hàm khởi tạo chứa 4 tham số là

Split\_attribute: là thuộc tính được chọn phân chia tại nút đó được tạo ra từ việc lựa chọn thuộc tính tốt nhất. ví dụ ‘outlook’

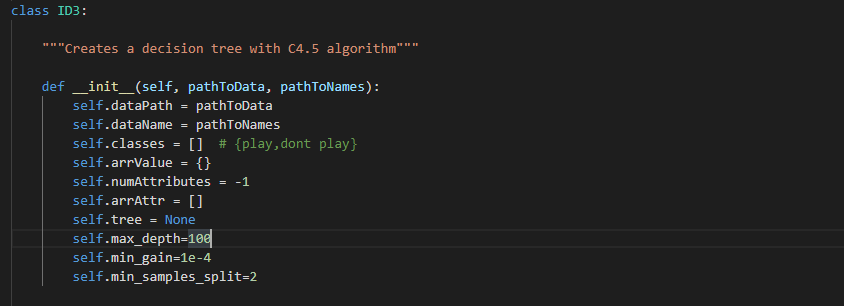
Ids: là danh sách các chỉ mục của các bản ghi ví dụ [ [1,2,3,4,5][6,7,8,9]] ví dụ [1,2,3,5,6] là chỉ mục của các bản ghi có chứa dữ liệu ‘Sunny’,[6,7,8,9] chứa dữ liệu ‘Overcast’ đối với thuộc tính split\_attribute

Children: tập các nút con của nó

Order: lưu tập các giá trị của thuộc tính split\_attribute [‘Overcast’,’outlook’,’rainny’]

Label: là nhãn của nút đó, nếu nút đó là lá sẽ được gãn tên

Threshold : là ngưỡng tốt nhất đối với thuộc tính split\_attribute (liên tục)



ID3 là tên cây:

Hàm khởi tạo gồm:

arrValue: lưu giá trị của các trường

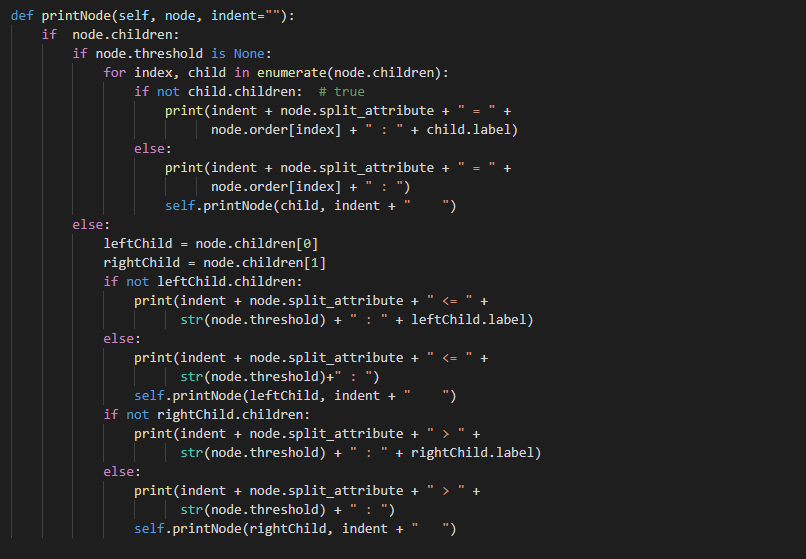
numAttributes số thuộc tính

max\_dept: độ sâu tối đa của cây

min\_gain: lượng thông tin tổi thiếu lựa chọn thuộc tính

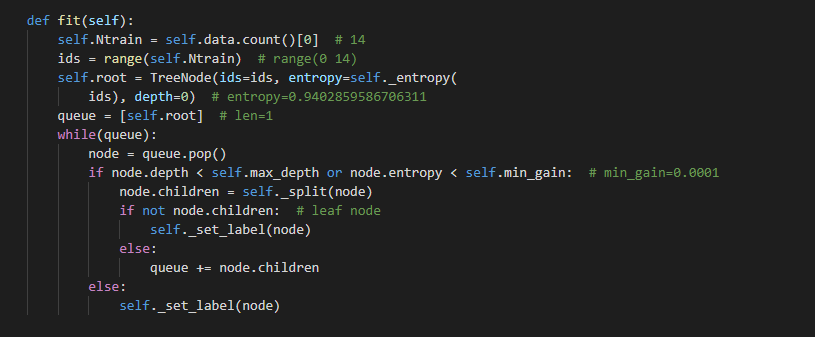
min\_samples\_splits: số bản ghi tối thiểu của của một nút

arrAtt: lưu tên tất cả các thuộc tính



Đầu vào hàm là một nút và một ký tự khoảng cách để hiển thị

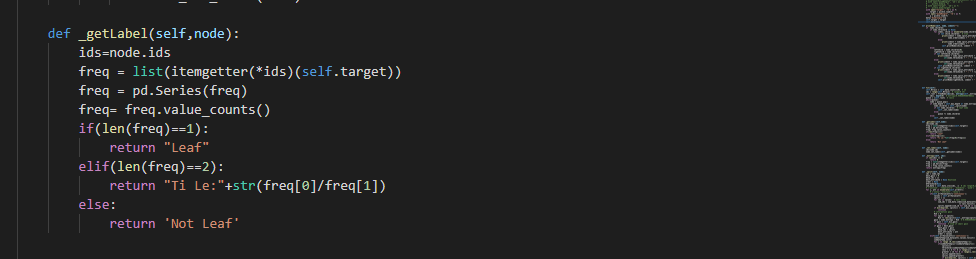
Hàm printNode là hàm đệ quy để hiện thị thông tin từng nút của cây quyết định đến nút lá



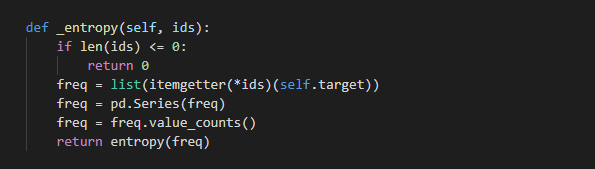
Hàm fit

Tạo ra một nút gốc và một hàng đợi để thêm các nút và lấy nút ra để chọn thuộc tính

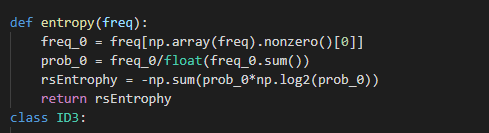
Nếu một nút được tạo ra là nút là sẽ được gán nhãn , còn không sẽ được thêm vào hàng đợi và tiếp tục chọn thuộc tính.



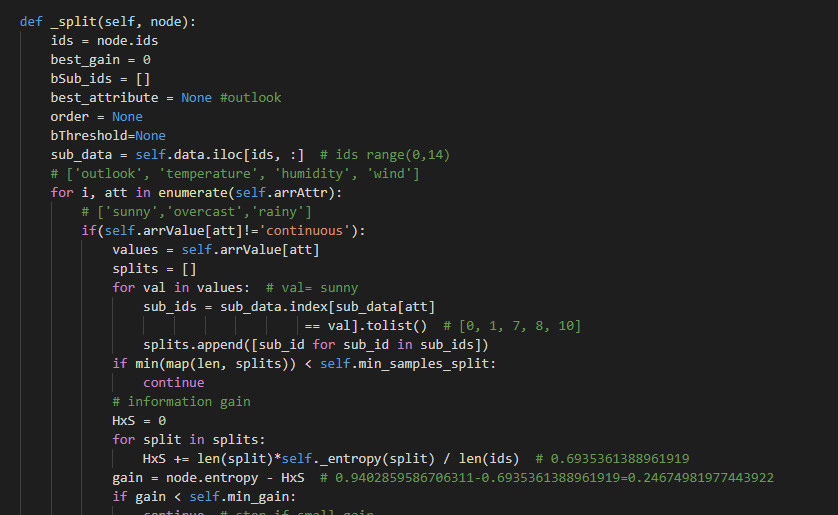
Hàm getLabel đễ kiểm tra xem nút đầu vào là nút lá hay không

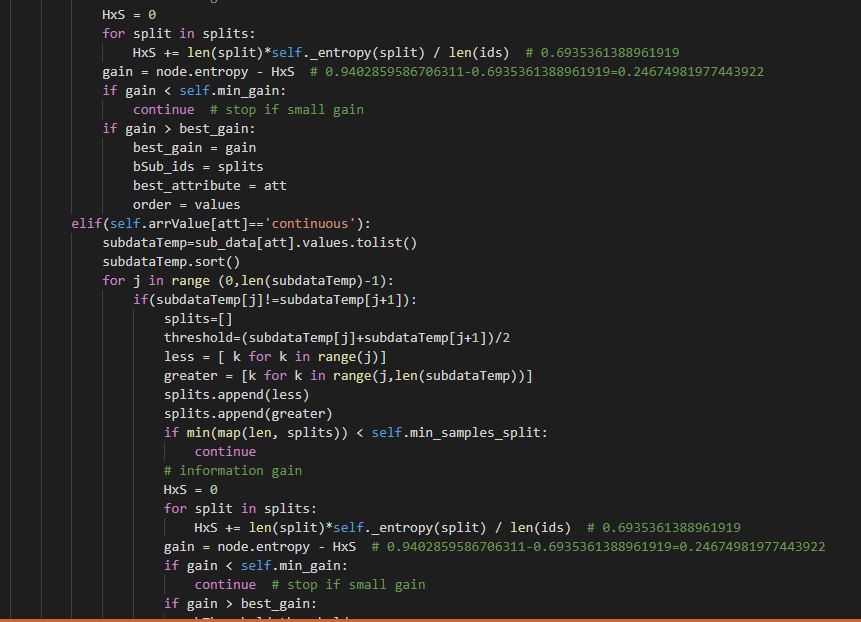


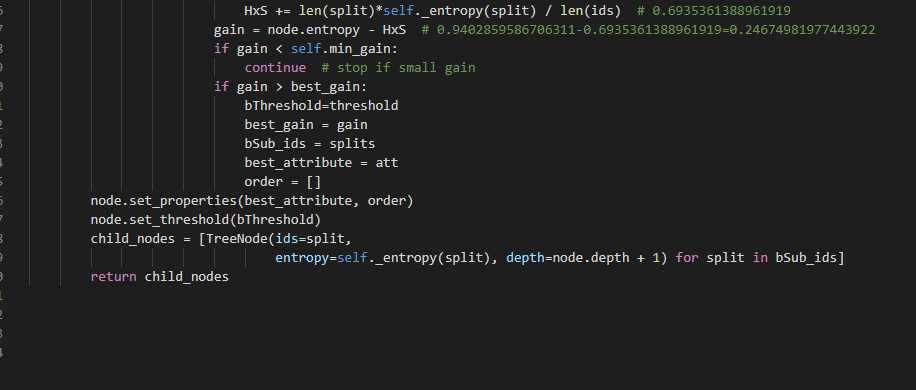
Hàm \_entropy: đầu vào làm một mảng chỉ mục của các dòng dữ liệu đầu ra là kết quả tính theo công thức entropy của mảng chỉ mục đó



Hàm entropy : để tính entropy







Hàm split:

Đầu vào: là một nút cần phân chia

Đầu ra: là danh sách nút con theo thuộc tính tốt nhất

Mục đích: Chọn thuộc tính tốt nhất để phân chia

Nguyên lý hoạt động: Lấy phần dữ liệu ứng với nút đầu vào, duyệt tất cả các thuộc tính cho dữ liệu này, đối với thuộc tính liên tục thì sắp xếp theo giá trị tăng dần rồi tìm ngưỡng để tính lượng thông tin thu được khi phân chia thành hai nút, đối với thuộc tính rời rạc tính lượng thông tin khi phân chia thành số nút ứng với số giá trị rời rạc. Sau đó chọn thuộc tính có lượng thông tin lớn nhất, trả về danh sách nút con ứng với thuộc tính đó.

    best\_gain = 0

        bSub\_ids = []

        best\_attribute = None #outlook

        order = None

        bThreshold=None

        sub\_data = self.data.iloc[ids, :]  # ids range(0,14)

các biến best\_gain lưu lượng thông tin lớn nhất

bSub\_ids: lưu các chỉ mục dữ liệu của thuộc tính tốt nhất

order: lưu giá trị của thuộc tính tốt nhật

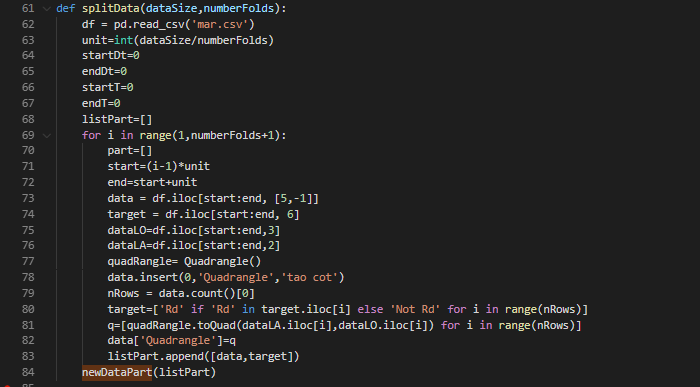
bThreshold: lưu ngưỡng của thuộc tính tốt nhất

sub\_data: lưu các dòng dữ liệu theo chỉ mục ids của nút đầu vào

đầu tiền hàm sẽ duyệt hết tất cả thuộc tính. Kiểm tra thuộc tính đố là kiểu liên tục hay rời rạc. Nếu là rời rạc sẽ tính entropy cho từng giá trị rời rạc. Nếu là liên tục sẽ tìm tất cả các ngưỡng giá trị để tính entropy

từ đó đầu ra sẽ là một nút tốt nhất

**Module sử dụng cross validation tìm ra mô hình tốt nhất.**



Hàm splitData:

Đầu vào: kích thước toàn bộ dữ liệu, tham số k cross validation

Đầu ra: Tệp lưu dữ liệu gồm k phần

Mục đích: Tiền xử lý dữ liệu đầu vào, tạo cột dẫn xuất Quadrangle, tạo cột dẫn phân lớp 2 giá trị Rd và Not Rd



Hàm createModelCrossValidation:

Đầu vào: Dữ liệu đã được chia thành nhiều phần

Đầu ra: Số mô hình tương ứng với số phần

Mục đích: Tạo ra k mô hình, để test và tìm ra mô hình có chỉ số chính xác tốt nhất và lưu các mô hình đó vào tệp.

# **Những kết luận rút ra, những thu hoạch của các thành viên trong nhóm (kiến thức, kỹ năng, ...)**

|  |  |  |
| --- | --- | --- |
| Tên Thành Viên | Kết luận | Thu hoạch |
| Nguyễn Nam Dương | Công cụ cây quyết định là một công cụ khai phá rất mạnh mẽ trên cả dữ liệu tương đối lớn | Hiểu hơn về cây quyết định  Hiểu hơn về sao Hỏa |
| Nguyễn Thu Huyền | Tìm hiểu các course book trong khóa học | - Các thuộc tính của miếng núi lửa  - Mô tả được các thuộc tính của miệng núi lửa |
| Nguyễn Thị Xuân | Tìm hiểu hai phương pháp đánh giá bias và trade off | - Khái niệm về Bias và đặc trưng của Variance  - Mối quan hệ của Bias và variance |
| Trần Thị Hải | Tìm hiểu các khái niệm cơ bản của học máy | - Khái niệm cơ bản về Machine learning  - Quy trình xây dựng machine leaning  - Phân loại |
| Dương Công Thành |  |  |