TRƯỜNG ĐẠI HỌC

SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH

HCMC University of Technology and Education

 KHOA ĐÀO TẠO CHẤT LƯỢNG CAO

NGÀNH CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN III

TÌM HIỂU BỘ PHÂN LOẠI

RANDOM FOREST

|  |  |
| --- | --- |
| Nhóm sinh viên thực hiện: | |
| Bùi Minh Trung  Nguyễn Hữu Huân | 17110243  17110146 |

GVHD: Th.s Từ Tuyết Hồng

Tp. Hồ Chí Minh, tháng 12 – 2020

ĐIỂM SỐ

|  |  |  |  |
| --- | --- | --- | --- |
| TIÊU CHÍ | NỘI DUNG | TRÌNH BÀY | TỔNG |
| ĐIỂM |  |  |  |

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Giáo viên hướng dẫn

(*ký và ghi họ tên*)

Từ Tuyết Hồng

LỜI CẢM ƠN

Để hoàn thành tốt đề tài và bài báo cáo này, chúng em xin gửi lời cảm ơn chân thành đến giảng viên, cô Từ Tuyết Hồng, người đã trực tiếp hỗ trợ chúng em trong suốt quá trình làm đề tài. Chúng em cảm thầy đã đưa ra những lời khuyên từ kinh nghiệm thực tiễn của mình để định hướng cho chúng em đi đúng với yêu cầu của đề tài đã chọn, luôn giải đáp thắc mắc và đưa ra những góp ý, chỉnh sửa kịp thời giúp chúng em khắc phục nhược điểm và hoàn thành tốt cũng như đúng thời hạn Khoa đã đề ra.

Chúng em cũng xin gửi lời cảm ơn chân thành các quý thầy cô trong khoa Đào tạo Chất Lượng Cao nói chung và ngành Công Nghệ Thông Tin nói riêng đã tận tình truyền đạt những kiến thức cần thiết giúp chúng em có nền tảng để làm nên đề tài này, đã tạo điều kiện để chúng em có thể tìm hiểu và thực hiện tốt đề tài. Cùng với đó, chúng em xin được gửi cảm ơn đến các bạn cùng khóa đã cung cấp nhiều thông tin và kiến thức hữu ích giúp chúng em có thể hoàn thiện hơn đề tài của mình.

Đề tài và bài báo cáo được chúng em thực hiện trong khoảng thời gian ngắn, với những kiến thức còn hạn chế cùng nhiều hạn chế khác về mặt kĩ thuật và kinh nghiệm trong việc thực hiện một dự án phần mềm. Do đó, trong quá trình làm nên đề tài có những thiếu sót là điều không thể tránh khỏi nên chúng em rất mong nhận được những ý kiến đóng góp quý báu của các quý thầy cô để kiến thức của chúng em được hoàn thiện hơn và chúng em có thể làm tốt hơn nữa trong những lần sau. Chúng em xin chân thành cảm ơn.

Cuối lời, chúng em kính chúc quý thầy, quý cô luôn dồi dào sức khỏe và thành công hơn nữa trong sự nghiệp trồng người. Một lần nữa chúng em xin chân thành cảm ơn.

TP.HCM, ngày tháng 12 năm 2020

Nhóm sinh viên thực hiện

MỤC LỤC

[Chương 1: Tổng quan chương trình 1](#_Toc58881789)

[1. Giới thiệu chung 1](#_Toc58881790)

[1.1. Về đồ án Tìm hiểu bộ phân loại Random Forest 1](#_Toc58881791)

[1.1.1. Yêu cầu đồ án 1](#_Toc58881792)

[1.1.2. Phân tích đồ án 1](#_Toc58881793)

[1.1.3. Phương hướng thực hiện 1](#_Toc58881794)

[1.2. Machine Learning 1](#_Toc58881795)

[1.2.1. Lí thuyết Machine Learning cơ bản 1](#_Toc58881796)

[1.2.2. Thư viện TensorFlow 2](#_Toc58881797)

[Chương 2: Thiết kế chương trình 3](#_Toc58881798)

[1. Dataset 3](#_Toc58881799)

[2. Thuật toán Random Forest 4](#_Toc58881800)

[2.1 Định nghĩa 4](#_Toc58881801)

[1.2. Hoạt động 4](#_Toc58881802)

[1.3. Ưu điểm 6](#_Toc58881803)

[1.4. Nhược điểm 6](#_Toc58881804)

[3. Code 6](#_Toc58881805)

[Chương 3: Kết luận và hướng phát triển 22](#_Toc58881806)

[1. Kết luận 22](#_Toc58881807)

[2. Hướng phát triển 22](#_Toc58881808)

[Tài liệu tham khảo 22](#_Toc58881809)

# Chương 1: Tổng quan chương trình

## Giới thiệu chung

### Về đồ án Tìm hiểu bộ phân loại Random Forest

#### Yêu cầu đồ án

Xây dựng chương trình xác định chất lượng xe hơi dựa trên các thuộc tính của xe bằng cách sử dụng thuật toán bộ phân loại Random Forest

#### Phân tích đồ án

* Xây dựng chương trình Machine Learning
* Sử dụng dataset để lấy dữ liệu đầu vào.
* Dữ liệu đầu ra đưa tới cho người dùng dạng biểu đồ.

#### Phương hướng thực hiện

* Sử dụng ngôn ngữ Python trên IDE visual studio 2019.
* Lấy Dataset từ Car Evaluation Data Set.

### Machine Learning

#### Lí thuyết Machine Learning cơ bản

Những năm gần đây, Artificial Intelligence (Trí Tuệ Nhân Tạo - AI), và cụ thể hơn là Machine Learning (Học Máy hoặc Máy Học) nổi lên như một bằng chứng của cuộc cách mạng công nghiệp lần thứ tư. AIđang len lỏi vào mọi lĩnh vực trong đời sống mà có thể chúng ta không nhận ra. Xe tự hành của Google và Tesla, hệ thống tự tag khuôn mặt trong ảnh của Facebook, trợ lý ảo Siri của Apple, trợ lí ảo Alexa của Amazon, Cortana của Microsoft, hệ thống gợi ý sản phẩm của Amazon, hệ thống gợi ý phim của Netflix, máy chơi cờ vây AlphaGo của Google DeepMind, …, chỉ là một vài trong vô vàn những ứng dụng của AI.

Machine Learning là một tập con của AI. Theo định nghĩa của Wikipedia, Machine learning is the subfield of computer science that “gives computers the ability to learn without being explicitly programmed”. Nói đơn giản, Machine Learning là một lĩnh vực nhỏ của Khoa Học Máy Tính, nó có khả năng tự học hỏi dựa trên dữ liệu đưa vào mà không cần phải được lập trình cụ thể.

#### Thư viện TensorFlow

TensorFlow là một thư viện Machine Learining được Google phát triển và phát hành vào tháng 10 năm 2015. Thư viện này hỗ trợ xây dựng các mô hình Machine Learning rất phức tạp qua những API cực kì ngắn gọn. Các mô hình Machine Learning phát triển trên TensorFlow có thể được sử dụng trên nhiều nền tảng khác nhau (từ Smartphone tới Distributed Servers) và trên cả CPUs lẫn GPUs.

# Chương 2: Thiết kế chương trình

## Dataset

Các cột đại diện cho CarEvaluation những điều sau:

* buying - giá xe
  + buying: 4 giá trị, ['vhigh' 'high' 'med' 'low']
* door- số cửa
  + doors: 4 giá trị, ['2' '3' '4' '5more']
* persons - số lượng hành khách cao nhất có thể được vận chuyển
  + persons: 3 giá trị, ['2' '4' 'more']
* lug\_boot - kích thước khoang hành lý / cốp
  + lug\_boot: 3 giá trị, ['small' 'med' 'big']
* safety - ước tính an toàn của ô tô
  + safety: 3 giá trị, ['low' 'med' 'high']
* class - khả năng chấp nhận xe hơi
  + class: 4 giá trị, ['unacc' 'acc' 'vgood' 'good']
* maint: độ bảo trì của xe
  + maint: 4 giá trị, ['vhigh' 'high' 'med' 'low']

Vấn đề:

* Quyết định mua một chiếc ô tô hay không tùy theo trình độ thể chất của nó đang được thảo luận trong trường hợp này. Dựa trên thông tin được cung cấp bởi tập dữ liệu, mỗi chiếc xe sẽ được phân loại, sử dụng sáu thuộc tính, thành không thể chấp nhận(unacc), chấp nhận được(acc), tốt(good) hoặc rất tốt(vgood).

Phương pháp:

* Một quá trình phân loại bằng cách sử dụng RandomForest được coi là chủ đề của câu chuyện này. Mô hình RandomForest là một phương pháp học có giám sát, các thuật toán học từ các nhãn của dữ liệu và một mô hình dự đoán hưởng lợi từ một số quy tắc nhị phân để tính toán giá trị của biến mong muốn.
* RandomForest tạo ra các mô hình phân loại ở dạng cây. Biểu mẫu này giúp hiểu hệ thống phân cấp quyết định và mối quan hệ giữa các thuộc tính bằng cách hình dung bằng cách sử dụng các kết quả có thể có của mỗi thuộc tính như một nhánh của cây. và nhanh hơn.
* Mặt khác, một thay đổi nhỏ trong dữ liệu có thể gây hại đáng kể cho cấu trúc cây cho các dự đoán. RandomForest dễ bị trang bị quá mức và chúng thường không thể đạt đến mức độ chính xác được cung cấp bởi các phương pháp phân loại thay thế. Tại thời điểm này, một mô hình quần thể rừng ngẫu nhiên được xây dựng để tăng mức độ chính xác trong nghiên cứu này.

## Thuật toán Random Forest

### . Định nghĩa

Random Forest là thuật toán học có giám sát (supervised learning). Nó có thể được sử dụng cho cả phân lớp và hồi quy. Nó cũng là thuật toán linh hoạt và dễ sử dụng nhất. Một khu rừng bao gồm cây cối. Người ta nói rằng càng có nhiều cây thì rừng càng mạnh. Random forests tạo ra cây quyết định trên các mẫu dữ liệu được chọn ngẫu nhiên, được dự đoán từ mỗi cây và chọn giải pháp tốt nhất bằng cách bỏ phiếu. Nó cũng cung cấp một chỉ báo khá tốt về tầm quan trọng của tính năng. Random forests có nhiều ứng dụng, chẳng hạn như công cụ đề xuất, phân loại hình ảnh và lựa chọn tính năng. Nó có thể được sử dụng để phân loại các ứng viên cho vay trung thành, xác định hoạt động gian lận và dự đoán các bệnh. Nó nằm ở cơ sở của thuật toán Boruta, chọn các tính năng quan trọng trong tập dữ liệu.

### Hoạt động

Chúng ta có thể nghĩ đến một ví dụ đơn giản trong cuộc sống, giả sử tôi muốn tìm hiểu một địa danh cho chuyến du lịch sắp tới, tôi sẽ đi hỏi một người bạn để tham khảo ý kiến. Nhưng, ý kiến của người bạn này có thể không khách quan cho lắm. Tôi liền đi hỏi thêm một vài người nữa, và tổng hợp lại để cho ra quyết định đi hay không.

Nếu coi mỗi ý kiến của những người góp ý là một cây quyết định, thì chúng ta đã có hình dung mơ hồ về Random Forest rồi.

Random Forest hoạt động bằng cách đánh giá nhiều Cây quyết định ngẫu nhiên, và lấy ra kết quả được đánh giá tốt nhất trong số kết quả trả về.

Mã giả cho hoạt động của Random Forest:

1. Chọn ngẫu nhiên “k” features từ tập “m” features.

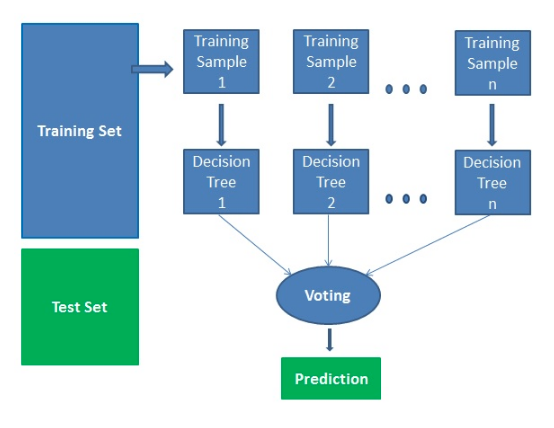
Để ý k << m

2. Từ tập “k” features, tính toán ra node “d” là tốt nhất cho Node phân loại.

3. Chia các node con theo node tốt nhất vừa tìm được

4. Lặp lại bước 1-3 cho đến khi đạt đến k node

5. Lặp lại bước 1-4 để tạo ra “n” cây



Sau các bước trên, chúng ta đã tạo ra được một Random Forest, vậy nó hoạt động như thế nào để dự đoán ?

RANDOM FOREST PREDICTION :

Để biểu diễn dự đoán sử dụng Random Forest đã huấn luyện, ta sử dụng các bước bên dưới :

1. Lấy các test features và sử dụng các Cây quyết định đã tạo ra để dự đoán kết quả, lưu nó vào một danh sách.

2. Tính toán số lượng vote trên toàn bộ Forest cho từng kết quả.

3. Lấy kết quả có số lượng vote lớn nhất làm kết quả cuối cho mô hình.

### Ưu điểm

* Random forests được coi là một phương pháp chính xác và mạnh mẽ vì số cây quyết định tham gia vào quá trình này.
* Không bị vấn đề overfitting.
* Thuật toán có thể được sử dụng trong cả hai vấn đề phân loại và hồi quy.
* Random forests cũng có thể xử lý các giá trị còn thiếu.
* Có thể nhận được tầm quan trọng của tính năng tương đối, giúp chọn các tính năng đóng góp nhiều nhất cho trình phân loại.

### Nhược điểm

* Random forests chậm tạo dự đoán bởi vì nó có nhiều cây quyết định.
* Bất cứ khi nào nó đưa ra dự đoán, tất cả các cây trong rừng phải đưa ra dự đoán cho cùng một đầu vào cho trước và sau đó thực hiện bỏ phiếu trên đó.
* Tốn thời gian.
* Mô hình khó hiểu hơn so với cây quyết định, nơi bạn có thể dễ dàng đưa ra quyết định bằng cách đi theo đường dẫn trong cây.

## 3. Code

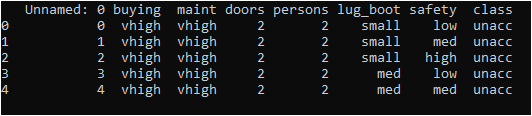


Sửa đổi dữ liệu và lưu nó trong tệp car\_data1.xlsx, để nó có thể được đọc dưới dạng excel.



Nhận thông tin về tập dữ liệu và kiểm tra các giá trị rỗng nếu có.

Kết quả:



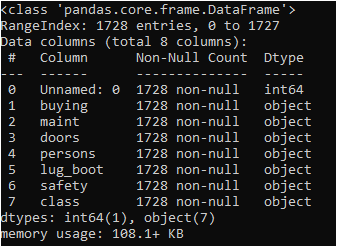


Thông tin dữ liệu.

Vì tất cả các cột đều được phân loại, hãy kiểm tra các giá trị duy nhất của mỗi cột.

Trong tập dữ liệu này có 1728 hàng dữ liệu không có mục nhập dữ liệu nào bị thiếu. Tất cả bảy cột hiện là kiểu dữ liệu đối tượng, về cơ bản là các chuỗi. Trong khi chuyển đổi các cột từ đối tượng Pandas sang kiểu dữ liệu danh mục Pandas sẽ tiết kiệm bộ nhớ (mặc dù nó tạo ra một chút khác biệt cho tập dữ liệu nhỏ này) và chỉ ra cho các thư viện Python khác kiểu dữ liệu thực, có rất ít lý do để làm như vậy vì dữ liệu sẽ được mã hóa để sử dụng trong mô hình học máy.

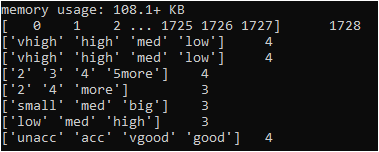
Kết quả:





Kiểm tra cách phân bổ các danh mục duy nhất này giữa các cột

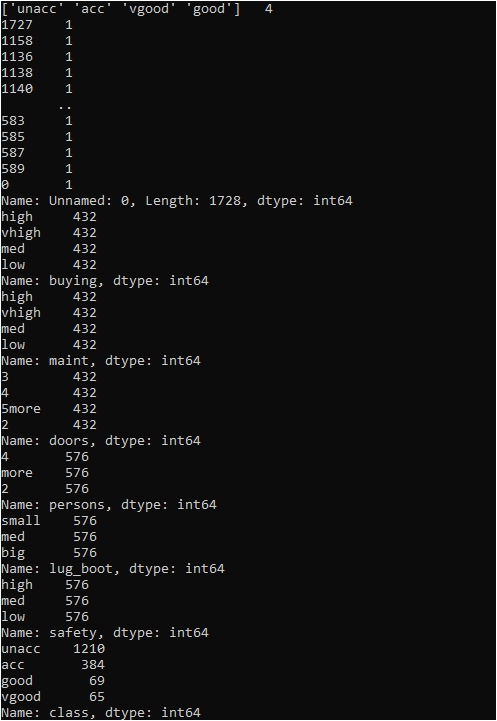
Kết quả:





Từ kết quả trên, rõ ràng là tất cả các cột ngoại trừ 'class' được phân phối như nhau giữa các dữ liệu. Dưới đây là biểu đồ cung cấp số lượng các giá trị duy nhất trong cột.

Kết quả:

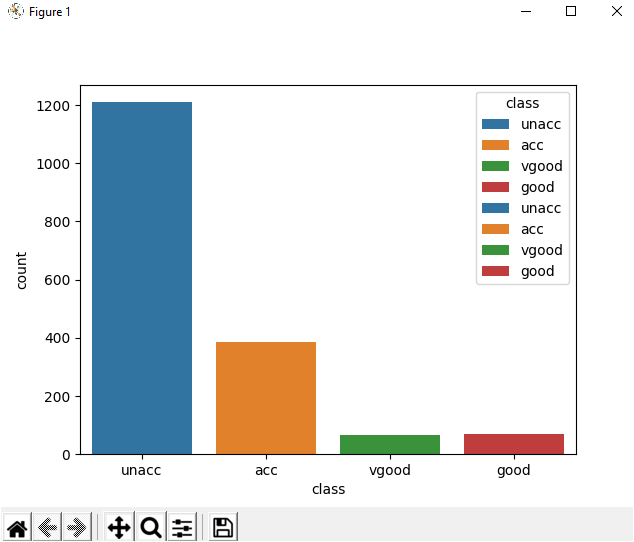


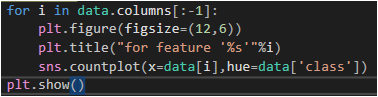


Có thể thấy từ biểu đồ rằng 'class' kết quả không cân bằng với các giá trị lớn hơn của 'unacc'. Vì vậy, đây là một bài toán phân loại đa thủy tinh không cân bằng.

Chúng ta đã kiểm tra cách phân phối 'class' cho từng tính năng trong dữ liệu của chúng ta.

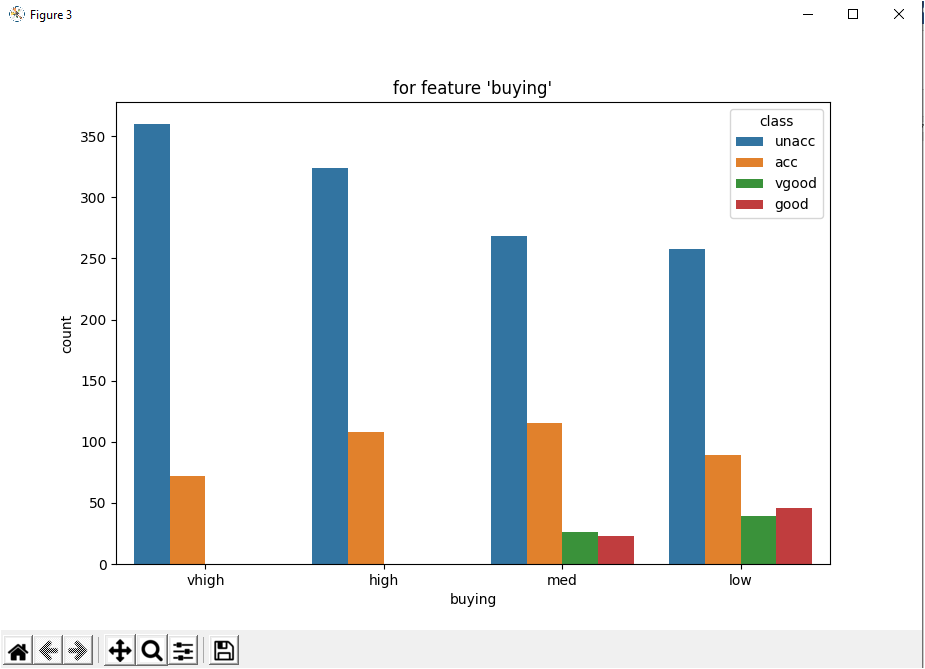
Kết quả:

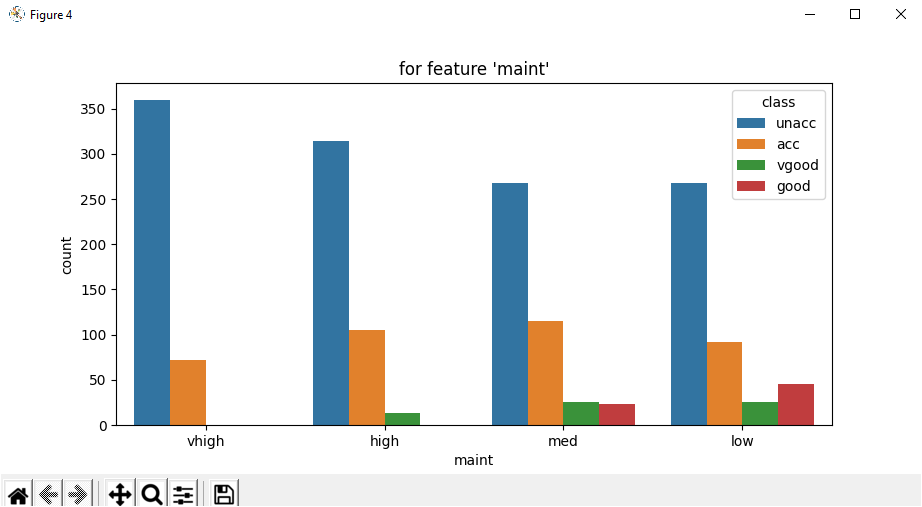


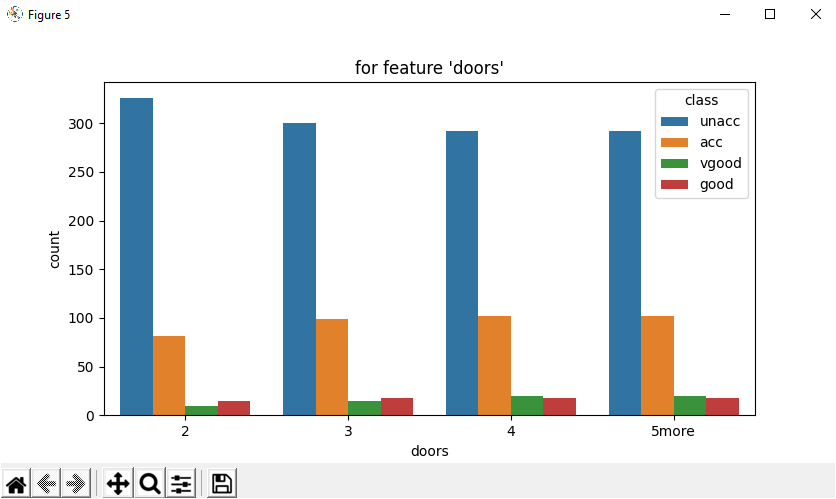


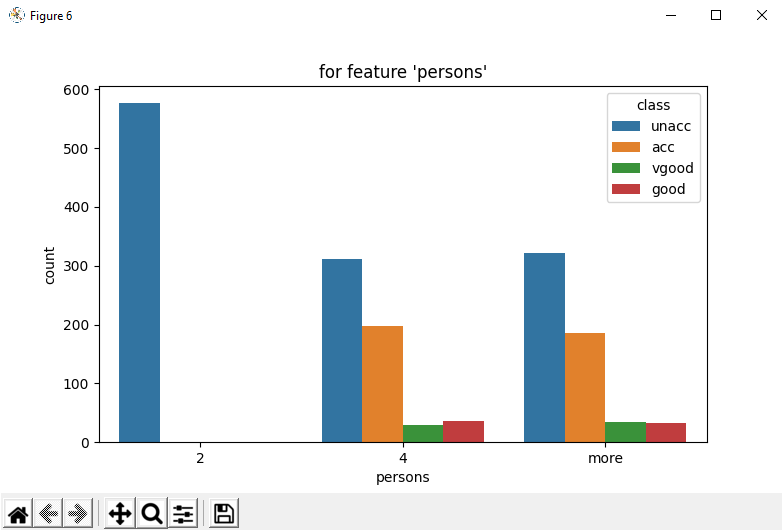
Vì các thuật toán scikit-learning thường không hoạt động với các giá trị chuỗi, nên chúng ta đã chuyển đổi danh mục chuỗi thành số nguyên.

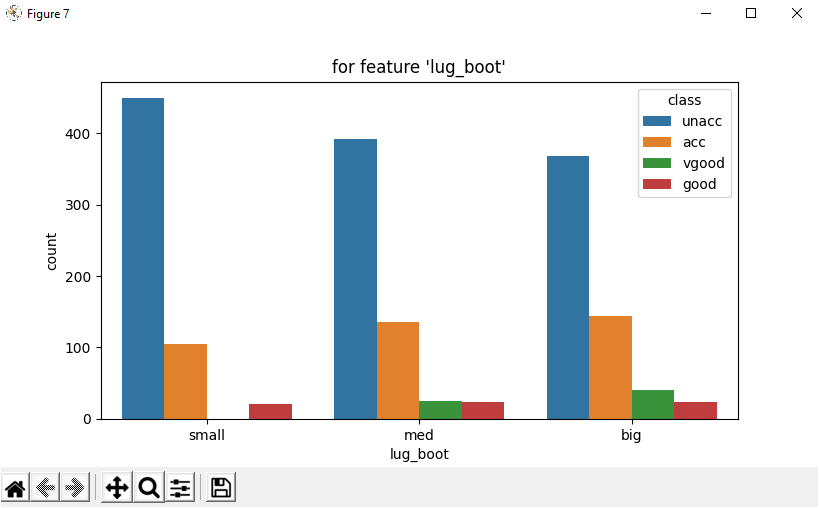
Kết quả:

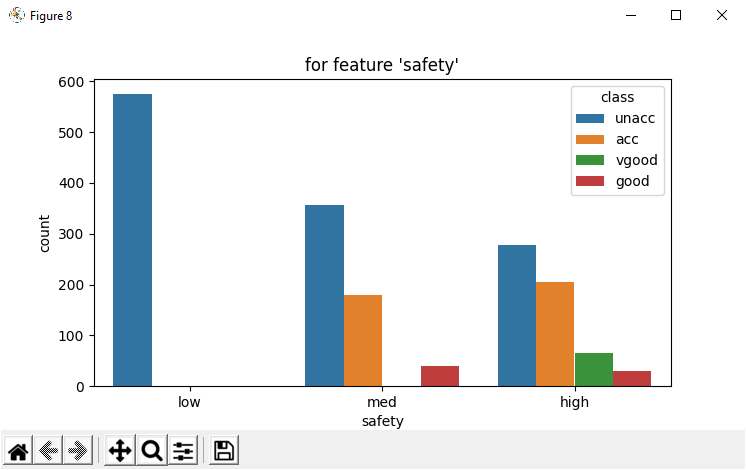


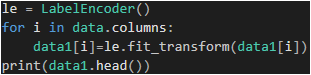








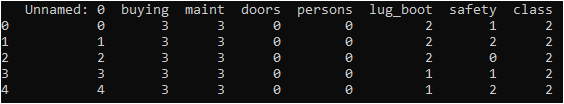




In dữ liệu đã chuyển đổi từ chuỗi thành số nguyên

Heatmap của các cột trên tập dữ liệu với nhau. Nó cho thấy hệ số tương quan của Pearson’s w.r.t các cột khác.

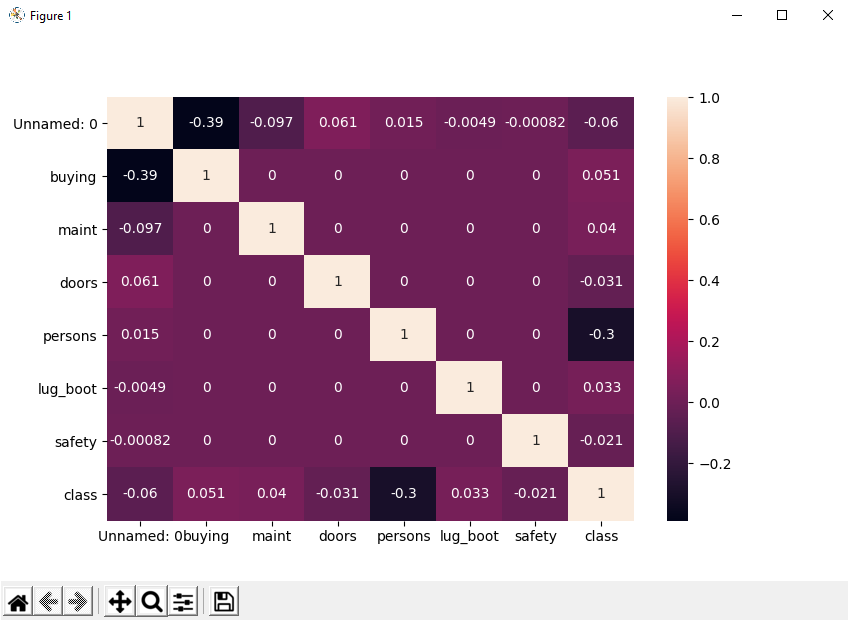
Kết quả:





Bỏ qua các giá trị đường chéo, có thể thấy rằng hầu hết các cột cho thấy mối tương quan rất yếu với 'class'. Cột 'people' đang hiển thị mối quan hệ yếu với 'class'. Các cột khác ngoại trừ 'class' không có mối tương quan nào với nhau. Vì vậy, việc vẽ các cột này với nhau hoặc thực hiện bất kỳ phân tích nào trên chúng có thể không mang lại bất kỳ sản lượng hiệu quả nào.

Kết quả:





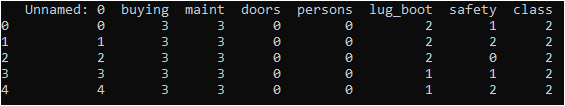


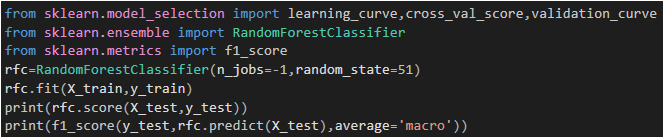
X là khung dữ liệu chứa dữ liệu / tính năng đầu vào

y là chuỗi có kết quả được dự đoán.

Phân chia dữ liệu trong các tập train và test set

Kết quả:





Cho Data học thuật toán RandomForestClassifier

Kết quả:



Vì vậy, mô hình cơ bản của RFC cho accuracy 96% và f1\_score 92%.

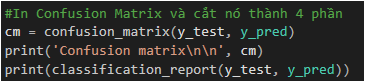


Dự đoán kết quả trên tập test

Ta được độ chính xác(accuracy) là 92%

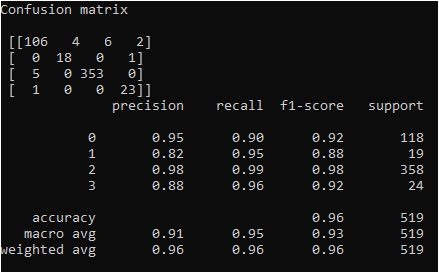
Kết quả:

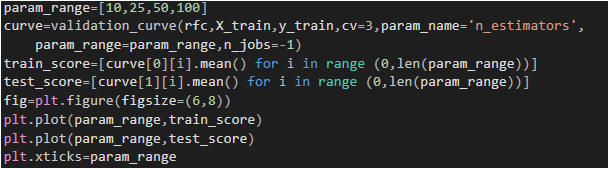




Báo cáo phân loại là một cách khác để đánh giá hoạt động của mô hình phân loại. Nó hiển thị độ chính xác(accuracy), precision, recall, f1 và điểm support cho model. Chúng ta đã mô tả các điều khoản này trong phần sau.

Kết quả:

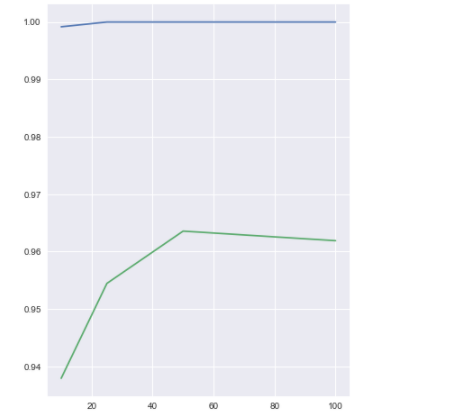


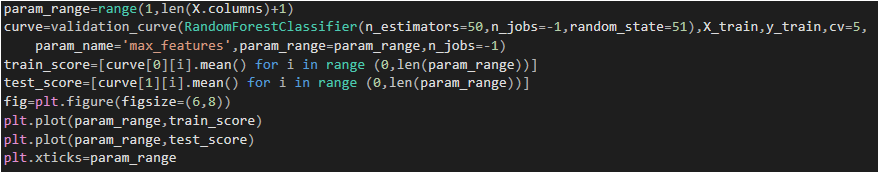


Bây giờ, hãy kiểm tra ảnh hưởng của n\_estimators đối với mô hình.

Vì vậy, với n\_estimators ngày càng tăng, accuracy của thử nghiệm ngày càng tăng. Mô hình đang đánh giá tốt nhất ở n\_estimators = 50. Sau n\_estimators = 50, mô hình bắt đầu quá mức. Bây giờ, chúng tôi đã đạt được khoảng. Accuracy 96,3%.

Kết quả:





Bây giờ, hãy kiểm tra cách mô hình phù hợp với các giá trị khác nhau của 'max\_features'.

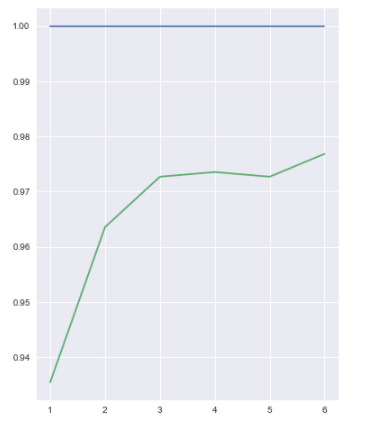
Từ biểu đồ trên, rõ ràng là mô hình đang mang lại kết quả tốt nhất cho max\_features = 6. Vẫn là model overfitting.

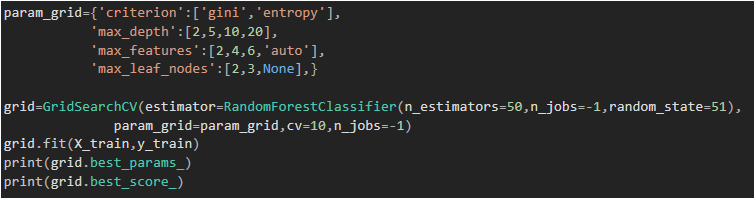
Bây giờ chúng tôi đã đạt accuracy 97,7%.

Chúng ta cũng có thể kiểm tra các tham số khác như 'max\_depth', 'tiêu chí', v.v. bằng cách sử dụng code trên.

Một cách đơn giản khác là sử dụng GridSearch để kết hợp các tham số tốt nhất. Vì dataset này nhỏ nên GridSearch sẽ mất ít thời gian hơn để hoàn thành.

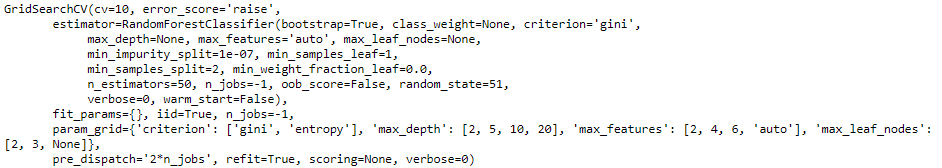
Kết quả:



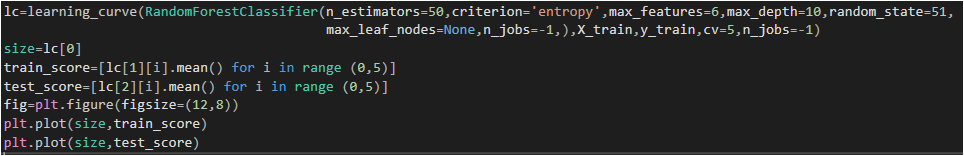


Vì vậy, với các thông số trên cho RFC, chúng ta đã đạt accuracy 98,42%.

Kết quả:

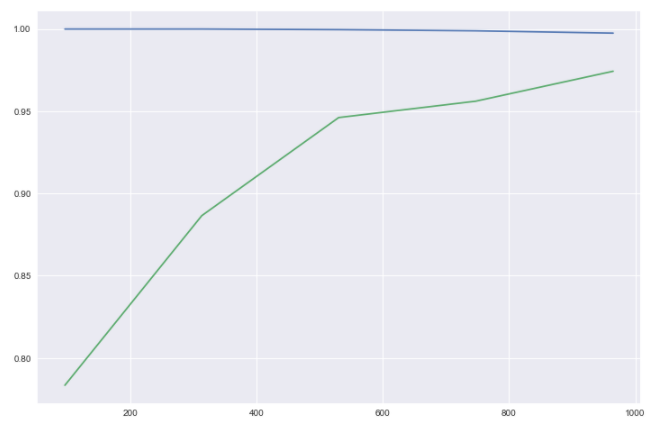




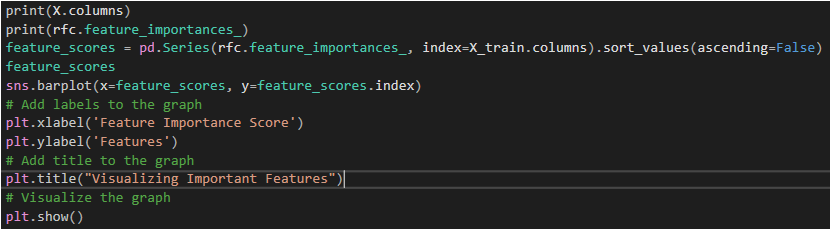


Learning Curve

Kết quả:

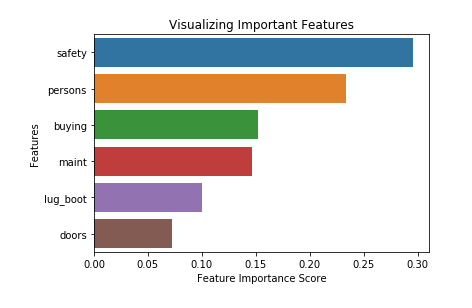


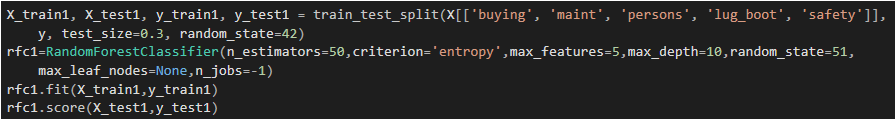
Model bị overfitting vì accuracy của tập train là 1, nhưng accuracy của tập test thấp hơn nhiều. Chúng ta đã thử thay đổi các thông số RFC để giải quyết overfitting. Nhưng, nó vẫn không giảm. Để giảm phương sai, chúng ta có thể Tăng số lượng samples. (Rõ ràng từ biểu đồ trên rằng tăng số lượng samples sẽ cải thiện mô hình) Giảm số lượng feature.



Từ các feature quan trọng, rõ ràng là feature 'doors' là ít quan trọng nhất và feature “safety” là quan trọng nhất. Vì vậy,train model của chúng ta loại trừ feature đó.

Kết quả:





Data của chúng ta đã có ít feature hơn và ngay cả khi chúng ta bỏ đi feature ít quan trọng nhất, thì accuracy cũng giảm xuống 93,64% Vì vậy, loại bỏ một feature không phải là một tùy chọn để giảm phương sai trong model của chúng ta. Lựa chọn duy nhất mà chúng ta còn lại là lấy thêm data.

Kết quả:



# Chương 3: Kết luận và hướng phát triển

## Kết luận

Về cơ bản, nhóm tự nhận xét phần mềm của nhóm đã giải quyết được được 90% yêu cầu mà đồ án đặt đặt ra. Sau đây là ưu điểm cũng như tồn tại nhược điểm của chương trình .

* Ưu điểm:
  + Chương trình đánh giá Data rõ ràng đơn giản bằng thuật toán phân loại RandomForest.
  + Chương trình tốn rất ít tài nguyên hệ thống khi hoạt động.
  + Chương trình chạy ổn định, cho ra kết quả chính xác, không bị crash trong quá trình thực thi yêu cầu người dùng.
* Nhược điểm:
  + Xử lí và xuất ra kết quả còn tốn nhiều thời gian.

## Hướng phát triển

* Tối ưu hoá thuật toán đối với dữ liệu đầu vào từ lớn đến rất lớn.

# Tài liệu tham khảo

[1] Car Evaluation Data Set

<https://archive.ics.uci.edu/ml/datasets/Car+Evaluation>

[2] Random Forest Classifier Tutorial

https://www.kaggle.com/prashant111/random-forest-classifier-tutorial