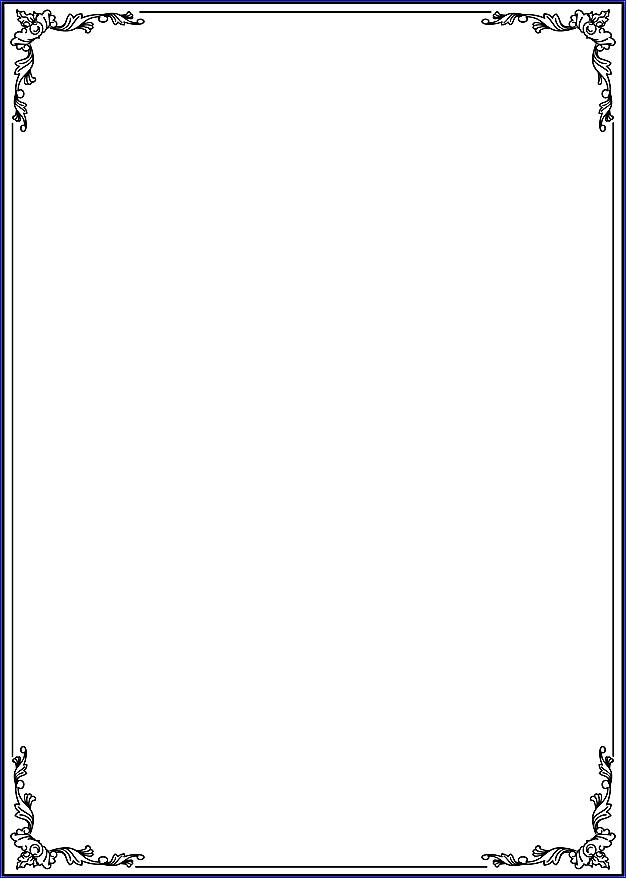
**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH**



**BÁO CÁO BẢO VỆ ĐỒ ÁN THỰC TẬP CHUYÊN MÔN**

**MÔN: THỰC TẬP CHUYÊN MÔN**

**Đề tài: PHÂN TÍCH MACHINE LEARNING CHO DỮ LIỆU BẢNG ÁP DỤNG KỸ THUẬT PHÂN TÍCH DIỄN GIẢI MÔ HÌNH THUẬT TOÁN TRONG Y HỌC**

**GVHD: Th.s Phạm Thị Miên**

**Th.s Nguyễn Văn Dũ**

***Nhóm sinh viên thực hiện:***

* Huỳnh Trọng Nhân MSV: 5951071066

TPHCM, ngày 10 tháng 06 năm 2021

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO THỰC TẬP CHUYÊN MÔN**

**ĐỀ TÀI: PHÂN TÍCH MACHINE LEARNING CHO DỮ LIỆU DẠNG BẢNG ÁP DỤNG KỸ THUẬT DIỄN GIẢI MÔ HÌNH THUẬT TOÁN VÀO VIỆC CHUẨN ĐOÁN BỆNH TRONG Y HỌC**

Sinh viên thực hiện: HUỲNH TRỌNG NHÂN

Lớp : CQ.59.CNTT

Khoá : **59**

TP. Hồ Chí Minh, năm 2021 tháng 06 ngày 15

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI **CỘNG HÒA XÃ HỘI CHỦ NGHIÃ VIỆT NAM**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH** Độc lập – Tự do – Hạnh phúc

# THIẾT KẾ TỔNG QUAN ĐỀ TÀI

**Mã sinh viên:**5951071066 **Họ tên SV:** Huỳnh Trọng Nhân

**Khóa:** 59 **Lớp:** CQ.59.CNTT

1. **Tên đề tài:**

Phân tích Machine Learning cho dữ liệu dạng bảng áp dụng kỹ thuật diễn giải mô hình thuật toán vào trong Y học

1. **Mục đích, yêu cầu:**

Giúp việc chuẩn đoán và phát hiện sẽ nhanh hơn quá trình xét nghiệm và chính xác hơn quá trình chuẩn đoán chủ quan của bác sĩ

1. **Nội dung và phạm vi đề tài:**

Gây tử vong gấp 5 lần so với bệnh cúm, COVID-19 gây ra tỷ lệ mắc và tử vong đáng kể. Gần giống như các bệnh tràn khí màng phổi khác, nhiễm trùng phổi với COVID-19 dẫn đến viêm và tràn dịch trong phổi. Hiện nay, COVID-19 chỉ có thể chuẩn đoán chính xác thông qua chuỗi phản ứng phức tạp Polymerase để phát hiện vật chất di truyền từ Virus. Tuy nhiên, có thể mất tới vài ngày (khoảng 7 ngày trong thời gian được cách ly) để có được kết quả xét nghiệm phần tử của bệnh nhân. Nhưng với việc quan sát các triệu chứng tôi có thể đánh giá và tiên lượng cho các bệnh nhân ở các mức độ khác nhau, nhưng có một điều cần lưu ý trong việc xây dựng mô hình chuẩn đoán bệnh trong y học đó là âm tính giả rất nguy hiểm và giải pháp sau có thể xử lý triệt để tình trạng âm tính giả để đem lại kết quả không quá tiêu cực cho một quốc gia phòng chống dịch

1. **Công nghệ, công cụ và ngôn ngữ lập trình:**

Sử dụng ngôn ngữ lập trình chủ yếu là R trong việc thống kê dữ liệu cho việc phân loại, dự đoán và Python trong việc làm cú pháp trở nên trực quan hơn. Ngoài ra còn sử dụng khá nhiều thư viện đi kèm trong việc xây dựng mô hình.

1. **Các kết quả chính dự kiến sẽ đạt được và ứng dụng:**

Mô hình có thể đạt được dự đoán chính xác là 65-85% cho việc chuẩn đoán bệnh nhân bị nhiễm bệnh và 75-80% cho việc tiên lượng tử vong của bệnh nhân. Mong muốn không xuất hiện các trường hợp âm tính giả ở cả 2 lần dự đoán.

1. **Giảng viên và cán bộ hướng dẫn**

Họ tên: Nguyễn Văn Dũ

Đơn vị công tác: Trường Đại Học Nông Lâm TP.HCM

Điện thoại: Email: [nguyenvandu.cpm@st.utc2.edu.vn](mailto:nguyenvandu.cpm@st.utc2.edu.vn)

Họ tên: Phạm Thị Miên

Đơn vị công tác: Trường Đại Học Giao Thông Vận Tải Cơ Sở 2

Điện thoại: 0961170638 Email:

|  |  |
| --- | --- |
|  | ***Ngày 04 tháng 06 năm 2021***  ***Giảng viên hướng dẫn*** |
|  |  |

**Table of Contents**

[**THIẾT KẾ TỔNG QUAN ĐỀ TÀI 3**](#_Toc76239200)

[**LỜI CẢM ƠN 8**](#_Toc76239201)

[**GIỚI THIỆU VẤN ĐỀ 10**](#_Toc76239202)

[**CHƯƠNG I: MÔ TẢ BÀI TOÁN VÀ GIẢI PHÁP 11**](#_Toc76239203)

[**1. Mô tả bài toán: 11**](#_Toc76239204)

[**1.2 Tại sao phải diễn giải mô hình 13**](#_Toc76239205)

[**1.3 Mục tiêu nghiên cứu. 15**](#_Toc76239206)

[**1.4 Vấn đề trong quy trình 15**](#_Toc76239207)

[**1.5 Giải pháp 16**](#_Toc76239208)

[**1.2.1 Phương pháp LIME 16**](#_Toc76239209)

[**1.2.2 Phương pháp RandomForestExplainer 18**](#_Toc76239210)

[**CHƯƠNG II: PHÂN TÍCH ĐÁNH GIÁ SƠ BỘ VỀ BỘ DỮ LIỆU VÀ XÂY DỰNG MÔ HÌNH 19**](#_Toc76239211)

[**2.1. Phân tích tổng quan về Dataset: 19**](#_Toc76239212)

[**2.2. Phân tích khám phá dữ liệu: 19**](#_Toc76239213)

[**2.3. Phần mềm và công cụ phát triển: 20**](#_Toc76239214)

[**2.4. Lựa chọn mô hình phát triển 21**](#_Toc76239215)

[**2.5 Tiến hành xây dựng mô hình: 24**](#_Toc76239216)

[**1. EDA 24**](#_Toc76239217)

[**2. Tiền xử lý: 34**](#_Toc76239218)

[**3. Dựng mô hình Random Forest: 36**](#_Toc76239219)

[**I. Mô hình chuẩn đoán bệnh 36**](#_Toc76239220)

[**II. Tiên Lượng tử vong 65**](#_Toc76239221)

[**CHƯƠNG III: KẾT LUẬN VÀ ĐÁNH GIÁ 87**](#_Toc76239222)

[**1. Đánh giá tổng quan. 87**](#_Toc76239223)

[**2. Ưu điểm và nhược điểm. 87**](#_Toc76239224)

[**2.1. Ưu điểm. 87**](#_Toc76239225)

[**2.2. Nhược điểm. 88**](#_Toc76239226)

[**3. Hướng phát triển. 88**](#_Toc76239227)

[**4. Lời Kết 91**](#_Toc76239228)

[**Tài liệu tham khảo 92**](#_Toc76239229)

# LỜI CẢM ƠN

Với lòng biết ơn sâu sắc nhất, em xin gửi đến quý Thầy Cô ở Khoa CNTT, Đăc biệt là thầy Nguyễn Văn Dũ và Cô Phạm Thị Miên đã truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường. Nhờ có những lời hướng dẫn, dạy bảo của các thầy cô nên đề tài nghiên cứu của em mới có thể hoàn thiện tốt đẹp.

Một lần nữa, em xin chân thành cảm ơn thầy/cô – người đã trực tiếp giúp đỡ, quan tâm, hướng dẫn em hoàn thành tốt bài báo cáo này trong thời gian qua.

Bài báo cáo thực tập thực hiện trong khoảng thời gian gần 5 tuần. Bước đầu đi vào thực tế của em còn hạn chế và còn nhiều bỡ ngỡ nên không tránh khỏi những thiếu sót , em rất mong nhận được những ý kiến đóng góp quý báu của quý Thầy Cô để kiến thức của em trong lĩnh vực này được hoàn thiện hơn đồng thời có điều kiện bổ sung, nâng cao ý thức của mình.

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

|  |
| --- |
| ***Tp. Hồ Chí Minh, ngày 09 tháng 06 năm 2021***  ***Giảng viên hướng dẫn*** |
|  |

# GIỚI THIỆU VẤN ĐỀ

Trong y học, khả năng giải nghĩa được cơ chế, nội dung của mô hình học máy là một trong những khả năng quan trọng quyết định tính khả dụng của mô hình đó, không phải ngẫu nhiên mà sau hành chục năm phát triển của thống kê ứng dụng, những bác sĩ lâm sàng vẫn đang ưa chuộng các thang điểm lâm sàng.

Lấy ví dụ các bác sĩ luôn sẽ áp dụng điểm [Apgar](file:///C:\Users\Hendrich\AppData\Roaming\Microsoft\Word\DocForReport\Apgar.rtf) (thang điểm đánh giá đầu đời) và điểm [Glasgow](file:///C:\Users\Hendrich\AppData\Roaming\Microsoft\Word\DocForReport\Glasgow.rtf) (thang điểm gây mê) khi tiếp xúc với bệnh nhân thay vì một mô hình Logistics hoặc mô hình cây, cho dù tình huống của bệnh vô cùng đơn giản, mô hình cây hoàn toàn có thể đáp ứng đủ điều kiện và bệnh viện không thiếu máy tính để chạy những mô hình phức tạp. Nhưng bởi vì, khi nhẩm tính thang điểm Glassgow, bác sĩ cấp cứu đang đồng thời quan sát về triệu chứng và dùng tư duy để suy luận về mức độ nghiêm trọng của tình trạng bệnh nhân. Nếu người ta tạo ra một Algorithm trong 1 chiếc camera thì bác sĩ sẽ ngưng suy luận và lệ thuộc vào cái máy này. Và điều này sẽ có sẽ dẫn tình huống phi logic trong quá trình chuẩn đoán bệnh

Tuy nhiên, mô hình chính dĩ nhiên tốt nhưng nó chưa vẫn chưa đủ. Độ chính xác của mô hình là tiêu chí hầu hết được nhắm tới bởi các data scientist, tuy nhiên, tính chính xác không hẵn trong trường hợp nào cũng đồng nghĩa với tính hiệu quả . Đặc biệt khi áp dụng vào y học, Machine Learning sẽ bị chi phối bởi rất nhiều thứ nhưng trung tâm vẫn là những quy tắc về Y đức và Lương tâm.

# CHƯƠNG I: MÔ TẢ BÀI TOÁN VÀ GIẢI PHÁP

## **Mô tả bài toán:**

Quay trở lại bài toán trên, tôi có thể đưa ra 3 trường hợp giả định như sau để làm rõ sự quan trọng trong việc diễn giải mô hình thuật toán:

Ta có một bệnh viện đang ra sức xây dựng mô hình tiên lượng tử vong cho bệnh nhân bị tai biến mạch máu não, sử dụng Machine Learning họ tạo ra một mô hình RandomForest cực kì chính xác, nó dự báo đúng kết quả **Tử Vong / Sống Sót** của bệnh nhân đến 98%. Trên 100 trường hợp chỉ sai lầm là 2. Tuy nhiên khi áp dụng mô hình trên lâm sàng, một số bác sĩ bắt đầu đặt ra những câu hỏi hoài nghi:

1. Tại sao bệnh nhân có nhiều tiền sử về bệnh tim mạch, huyết áp, thậm chí là hô hấp mãn tính thì mô hình lại tiên lượng là người có tỷ lệ sống sót cao, trong khi một bệnh nhân trẻ tuổi, hoàn toàn khỏe mạnh thì lại nhận tiên lượng tử vong rất cao ?
2. Tại sao người thành phố và cư ngụ tại các quận trung tâm lại có tiên lượng sống sót cao, còn người ở các vùng nông thôn, xa thành phố lại nhân kết quả tiên lượng tử vong ?

Bắt đầu có những lời đồn trong bệnh viên về sự phân biệt giữa các bệnh nhân giàu và nghèo, người thành phố và nông thôn.

Sau khi ngồi lại thảo luận giữa các chuyên viên thống kê và các bác sĩ, họ đã nhận ra một điều khá bất ngờ:

1. Mô hình đã phản ánh chính xác hoàn toàn thực tế: những bệnh nhân có tiền xử về các bệnh lý và các yếu tố nguy cơ cao thường được chăm sóc một các đặc biết. khảo sát một vòng quanh bệnh viện họ thấy được rằng một giường bệnh của một bệnh nhân như vậy cứ xấp xỉ 5 phút họ sẽ được một y tá viên tiến hành kiểm tra huyết áp đông thời cũng xấp xĩ 2 – 2 tiếng rưỡi sẽ có một bác sỹ tiến hành đánh giá sơ bộ tình trạng bệnh nhân, nhưng quay lại với những bệnh nhân trẻ tuổi khảo sát cũng chỉ ra rằng trong một khu vực nghiên cứu 100 bệnh nhân có đến 89 số bệnh nhân từ chối sự chăm sóc kiểm tra liên tục của các y tá và bác sĩ với lý do là cơ thể họ khỏe mạnh, cảm thấy bất tiện trong quá trình chăm sóc liên tục như vậy. Có thể nhìn thấy được việc tiên lượng trên của mô hình là hoàn toàn có cơ sở việc từ chối chăm sóc và kiểm tra liên tục của các cán bộ y tế khiến tiên lượng Sống sót của họ giảm đi một cách đáng kể, đáng nói đồng thời phần tiên lượng này lại tăng mạnh đối với người bệnh được chăm sóc liên tục

(*trích Survey of Medical Care For People – University Hospital Zurich*)

1. Trong tập dữ liệu của thông tin bệnh nhân có một features khá quan trọng đó là địa chỉ cư trú đồng thời trong hồ sơ bệnh án của họ cũng có một crittial feature là độ trễ trong cấp cứu lấy thí dụ: trong trường hợp 2 bệnh nhân đều ngã và được người nhà phát hiện với cùng thời gian tiệm cận là K, trong trường hợp này độ trễ trong cấp cứu được quyết định bởi khoảng cách từ địa chỉ cư trú đến bệnh viện như thế features này sẽ trở thành một yếu tố quyết định sự sống của bệnh nhân và tôi có công thức:

Với K là thời gian từ lúc bệnh nhân đột quy đến lúc phát hiện

Q là khoảng cách từ nhà đến bệnh viện

V là vận tốc tối đa cho việc vận chuyển bệnh nhân trong này tôi cho nó là hằng số = 60

F là thời gian nhập viện, giao động từ 180 – 360. Trường hợp xấu nhấy F = 360 khi ấy nếu O(t) > 0 tỷ lệ bệnh nhân tử vong sẽ cao  
nếu O(t) < 0 tỷ lệ cứu sống bệnh nhân cao. Suy xét trong những trường biến thiên còn lại tỷ lệ bệnh nhân được cứu sống là rất cao.

Ta giả định nếu tin đồn như trên được đồn thổi tới báo chí. Có lẽ nó sẽ dấu chấm hết cho việc áp dụng công nghệ vào trong y học chuẩn đoán và người thực hiện nó sẽ thành người thực hiện những thí nghiệm vô nhân đạo. Bây giờ chắc mọi người cũng đã biết lý do tạo sao tính chính xác không phải là tất cả. Lấy ví dụ như một cái ao nước rất đục một người bạn nhảy xuống và nói rằng nước rất cạn bạn có tin không rõ ràng đáp án về mặt lý luận chắc chắn là không bởi vì nó không tường minh không gì chứng minh được nước cạn. Như vậy một mô hình khi không thỏa được tính tường minh của mô hình đó sẽ trở nên vô nghĩa thậm chí là nguy hiểm

## **Tại sao phải diễn giải mô hình**

Thời gian gần đây bắt đầu có những ý kiến trái chiều, hoài nghi, thậm chí cảnh báo về phong trào sử dụng Big Data và Machine Learning vào trong y học. Một trong những luận điểm mà các bác sĩ lâm sàng đưa ra, đó là hầu hết những mô hình có nội dung không thể giải thích được. Khi họ không hiểu rõ được cơ chế đằng sau kết quả, thì họ không thể tin cậy vào bản thân kết quả đó. Kết quả đôi khi ảnh hưởng rất quan trọng tới bệnh đặc biệt trên phương diện tinh thần hoặc lựa chọn phương thức thực hiện xử lý bệnh.

Lấy ví dụ một khối u lành tính nhưng qua dự đoán của mô hình cho ra kết quả dự đoán Dương tính giả với khối u thuộc loại điển hình với 3 hiện tượng là viêm phổi nhiễm trùng Pneumocystis, xuất huyết phế nang và u hạt ác tính chỉ vì bệnh nhân có hiện tượng sử dụng thuốc lá. Như vậy đầu tiên kết quả đã ảnh hưởng một cách tiêu cực đến bệnh nhân mặc dù chỉ là dương tính giả nhưng phần nào kiến bệnh tình hiện tại người bệnh có chiều chuyển biến một cách xấu đi, thứ hai nếu dựa vào kết quả sai lệch đó các bác sĩ sẽ phải bàn bạc cùng với bệnh nhân về phương thức điều trị là lựa chọn phẩu thuật cắt bỏ khối u hay sử dụng phương thức hóa trị và xạ trị nội khoa. Như vậy bằng một cách kiểm chứng không xác thực và sai lầm của hình đã thay đổi trực tiếp cuộc sống của một bệnh nhân

Khác với Việt Nam, bệnh nhân tại nước ngoài có quyến được biết lý do cũng như lịch trình điều trị của họ và bác sĩ không thể giải thích rằng: “Ông / Bà có 99,5% bị ung thư tuyến giáp và phải tiến hành phẩu thuật cắt bỏ nó, vì một mô hình Neural đã quyết định như vậy”. Không bàn tới chính xác, nếu việc đưa ra các lời khẳng định hay quyết định như vậy thì không thể làm bất cứ điều gì, việc dự đoán là của mô hình nhưng quyền quyết định lại thuộc về bác sĩ nên đây là lý do cần phải tường minh trong việc xử lý của mô hình để chứng năng tính khả dụng của nó trong y học nói riêng và trong đời sống nói chung.

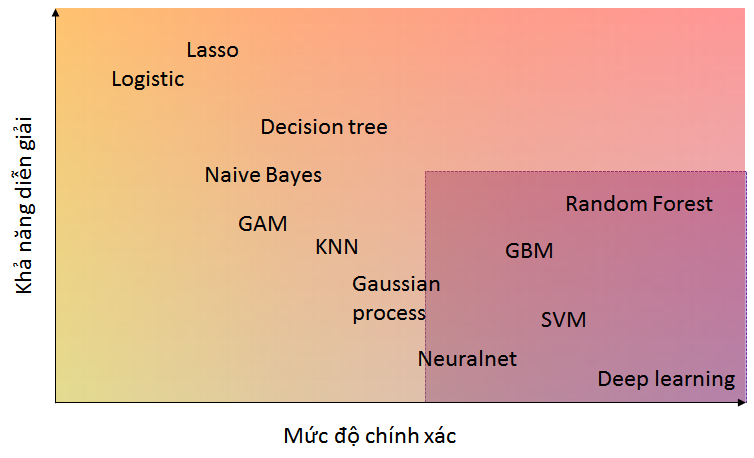
Vào năm 2018 tại Stuttgart, Germany có một điều luật bắt buộc mọi quyết định dựa vào Algorithm phải có khả năng lý giải nguyên nhân cho công dân có liên quan trực tiếp. Nếu luật này được thi hành ở Việt Nam thì, thì khách hàng có quyền yêu cầu ngân hàng giải thích tại sao hồ sơ xin vay vốn của họ bị xếp loại “Có nguy cơ cao”. Tương tự, bệnh nhân có thể tra hỏi bác sĩ tới cùng về kết quả CT của họ cho đến khi chính họ nhận được lời giải thích hợp lý , chính đáng

## **Mục tiêu nghiên cứu.**

Nghiên cứu nhằm đánh giá tổng quát về việc áp dụng thuật toán trong y học cũng như việc diễn giải tính tường minh của chúng giúp chúng có thể áp dụng một cách tối ưu hóa nhất trong y học. Ngoài ra, việc áp dụng mô hình thuật toán vào trong y học giúp tối đa hóa lợi ích, thời gian và nhân lực cho nền y học nước nhà

## **Vấn đề trong quy trình**

Trong machine Learning có một lời nguyên tai hại, đó là “Tính tường minh của một mô hình sẽ tỷ lệ nghịch với tính chính xác của nó”. Nghĩa là một hình càng chính xác thì càng phức tạp. Sự phức tạp đó đến từ bản chất của thuật toán, số lượng Input feature hoặc là cả 2. Nhưng trong thời đại Big Data và IOT đã cưỡng ép chúng ta phải dùng những thuật toán phức tạp nhất, vì những mô hình tuyến tính không có cách nào để xử lý được lượng lớn dữ liệu đó.



Trong hình trên, nhưng mô hình chính xác nhất lại là những mô hình bí hiểm nhất, bao gồm: RandomForest, Gradient Boosting, Deep neural network, Super Vector Machines… chúng trở thành những hộp đen bất khả tri.

* 1. **Giải pháp**

**1.2.1 Phương pháp LIME**

Năm 2016, một phương pháp có tên L.I.M.E đã được ra đời, nhằm diễn giải nội dung của các Black box này. Được viết tắt từ nguyên gốc của Local Interpretable Model-Agnostic Explanations. Tạm dịch là phương pháp diễn giải cục bộ cho mô hình bất khả tri.

Được giới thiệu lần đầu trong một bài báo của 3 Data Scientist tại ĐH Washinton là Marco Tulio Ribeiro, Sameer Singh và Carlos Guestrin. LIME có thể diễn giải bất cứ mô hình Classifier nào, bao gồm các mô hình Blackbox theo quy trình sau:

1. Phương pháp này dựa trên một giả định cơ bản đó là bất kể mô hình phức tạp đến đâu, thì tại một miền cục bộ trong không gian dữ liệu, mô hình có thể được ước lượng xấp xỉ bằng quy luật tuyến tính. Có thể hình dung trên một vài bệnh nhân có đặc tính input data xấp xỉ như nhau thì kết quả output của model sẽ như nhau. Đó là bằng chứng về quy luật tuyến tính cục bộ.
2. Đầu tiên LIME sẽ lấy thông tin về đặc tính phân phối của features (dữ liệu đầu vào) dựa vào training dataset và nội dung (feature nào được dùng) trong model. Thông tin này được lưu trữ trong một object gọi là explainer
3. Việc diễn giải sẽ áp dụng cho một trường hợp cá thể mới (unseen case), thí dụ bệnh nhân bất kỳ trích từ tập testcase. LIME sẽ mô phỏng một lượng lớn các trường hợp giả định nằm kề cận chung quanh trường hợp của điểm dữ liệu đang được xét
4. LIME áp dụng mô hình cho toàn bộ điểm trong không gian nhiễu này đồng tính khoảng cách giữa các điểm mô phỏng đến các điểm chung tâm (trường hợp đang được xét). Khoảng cách này sẽ được chuyển thành thang điểm.
5. Chọn ra số lượng cột features tiêu biếu (có khả năng gây nhiễu cao) cho phép mô tả tốt nhất khoảng cách nói trên
6. Dựng một mô hình đơn giản cho các điểm mô phỏng sử dụng m features được chọn làm predictor, để giải nghĩa outcome của model
7. Việc diễn giải tính hợp lý của kết quả được thực hiện dựa vào Weight coefficient và danh sách M features được chọn, Nếu Wc > 0, thì giá trị quan sát của features đang ủng hộ kết quả tiên lượng outcome của mô hình, vậy ngược lại thì sao? Nếu Wc < 0 thì giá trị features đang xung đột hay nói các khác là đang chống lại kết quả của dự đoán của mô hình

**1.2.2 Phương pháp RandomForestExplainer**

Đây là phương pháp giải nghĩa mô hình một cách tường tận và chi tiết thông qua các biểu đồ biểu thị số liệu, là một phương pháp cực kỳ thích hợp trong việc diễn giải các vấn đề trong y học một nơi cần thông tin minh bạch với độ chính xác đạt ngưỡng tối đa. Trong báo cáo phương pháp này được sử dụng theo quy trình như sau:

1. Khởi tạo mô hình với số cây và M feature cần thiết của dữ liệu hiển thị biểu đồ đường cong học tập của mô hình, quan sát đánh giá chất lương của mô hình qua biếu đồ.
2. Sau khi có được chất lượng của mô hình, tiến hành khoanh vùng các triệu chứng ảnh hưởng đến outcome của mô hình và thể hiện nó ra biểu đồ từ đó ra những bước đầu đánh giá sơ bộ về mô hình
3. Tiến hành thể hiện các features có ảnh hưởng đến mô hình nhất. Từ đó nghiên cứu về các tắc động của từng feature như là số cây, tổng số nút cho features đó, độ sâu tối thiểu … Tiếp theo thể hiện nó vào char cho từng feature và rút ra những đánh giá tương quan về các tác động của features vào outcome của model
4. Sử dụng mô hình phân bổ sự phân bố của các feature trong việc predict của mô hình tiến hành bổ sung việc đáng giá toàn bộ thể trạng bệnh nhân và thảo luận giữa các chuyên gia y tế đưa ra quyết định điều trị cho bệnh nhân về bệnh của họ

# CHƯƠNG II: PHÂN TÍCH ĐÁNH GIÁ SƠ BỘ VỀ BỘ DỮ LIỆU VÀ XÂY DỰNG MÔ HÌNH

1. **Phân tích tổng quan về Dataset:**

Bộ dữ liệu được sử dụng trong bài báo cáo gồm có 2 loại tương ứng với từng chức năng mà nó thực hiện:

1. [Wuhan.csv](file:///C:\Users\Hendrich\AppData\Roaming\Microsoft\Word\DataSet\WuhanVirus.csv) đây là bộ dữ liệu được trích xuất từ tập dữ liệu chính với hơn 364.251 hàng và 24 cột. Vì là trích xuất nên đây là cơ sở dữ liệu giản lược chỉ bao gồm 12 cột và 1.000 hàng. Bộ dữ liệu thể hiện dữ liệu của các bệnh nhân thông qua các triệu chứng phổ biến và đặc trưng từ đó phân lớp từng loại bệnh. (\*)
2. [DeathCovid.arff](file:///C:\Users\Hendrich\AppData\Roaming\Microsoft\Word\DataSet\ThoraricSurgery.arff) là bộ dữ liệu tiên lượng bệnh nhân xác xuất bệnh nhân tử vong dựa trên những triệu chứng của trong khoảng thời gian sinh hoạt thường nhật và trong lúc lâm sàng của bệnh nhân. Từ đó đưa ra giả thuyết dự đoán có cơ sở về tính mạnh của bệnh nhân (\*)
3. **Phân tích khám phá dữ liệu:**

“*The best preparation for tomorrow is doing your best today*”

Trước khi xây dựng mô hình bạn cần xây dựng một đặc trựng. Nhưng để làm được điều đó bạn cần phải biết khám phá dữ liệu.

Khám phá dữ liệu hay gọi tắt là EDA sẽ giúp chúng ta có một cái nhìn đầu tiên về dữ liệu, có được một cảm giác nhất định về những gì mình có trong tay trước khi có những chiến lược xây dựng mô hình. EDA giúp ta mườn tượng được độ phức tạp của bài toán và vạch ra những bước đầu tiên cần làm.

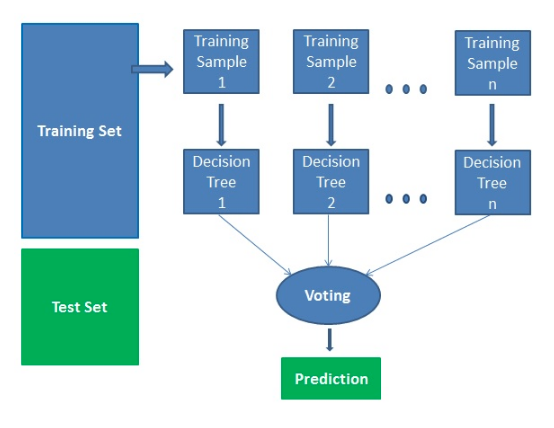
Việc khám phá dữ liệu không chỉ dừng lại ở lần đầu tiên trước khi xây dựng mô hình mà còn trược thực hiện trong suốt quá trình phát triển hệ thống. Sau khi xây dựng xong các đặc trưng, tôi cũng cần làm lại EDA một lần nữa để xem dữ liệu đã qua những thao tác xử lý đó đã thực sự sạch chưa. Ngoài ra, sau khi xây dựng và phân tích mô hình, tôi cũng cần thường xuyên quay lại EDA để tiếp tục khám phá điều ẩn giấu trong dữ liệu của bài toán. Càng hiểu sâu về dữ liệu, ta sẽ càng sớm giải thích được những hành vi của mô hình và đưa ra những thay đổi phù hợp.

Trong cả 2 tập dữ liệu được sử dụng trong đồ án này toàn bộ đều được EDA một cách kỹ càng sẽ không xuất hiện bất cứ hiện tượng dữ liệu nào bị khuyết nhưng yếu điểm ở đây xuất hiện hầu hết ở cả 2 là tập dữ liệu bị nhiễu và trong dữ liệu này không xuất hiện bất cứ features nào dư thừa tất cả các features đều có thể dùng cho mục đích chuẩn đoán nên đây là dữ liệu đã được làm sạch.

1. **Phần mềm và công cụ phát triển:**

* Rstudio và RCRAN đây là bộ IDE dành riêng cho lập trình viên R. Với việc hỗ trợ tối đa trong việc lập trình thống kê như biểu thị nôi trường tham số đang thực thi, thực thi từ câu lệnh riêng lẻ giống notebook, việc cài đặt thư viện khá dễ dàng và mô tả biểu đồ trong IDE giúp cho việc thay đổi tham số trong code tiện lợi hơn
* Azure Machine Learning có thể ví như đây giải pháp cho máy không còn nhiều dung lượng vì nó là công cụ giao diện người dùng đồ họa hỗ trợ xây dựng và triển khai các mô hình Machine Learning trên đám mây của Microsoft
* Môt số thư viện cần thiết cho việc triển khai mô hình được khai báo và trích dẫn tính năng cũng như công dụng trong một Document riêng [ở đây](file:///C:\Users\Hendrich\AppData\Roaming\Microsoft\Word\R\lib.docx)

1. **Lựa chọn mô hình phát triển**



**Mô tả:**

* Đây được coi như là một tập hợp mô hình (ensemble) có hiệu quả nhất theo cái nhìn chủ quan từ phía tôi
* Mô hình này được cấu tạo từ một thuật toán riêng biệt khác. Đó là cây quyết định.
* Mô hình Random Forest rất hiệu quả cho các bài toán phân loại vì nó huy động cùng lúc hàng trăm tới hàng nghìn mô hình nhỏ hơn bên trong với quy luật vận hành khác nhau để đưa ra quyết định cuối cùng. Mỗi mô hình có thể mạnh yếu khác nhau, nhưng theo nguyên tắc “[Wisdom of the Crowd](file:///C:\Users\Hendrich\AppData\Roaming\Microsoft\Word\Doc\Category\The%20Wisdom%20of%20Crowds.docx)”, tôi sẽ cơ hội phân loại chính xác hơn so với bất kỳ một mô hình đơn lẻ nào trong điều kiện phân phối dữ liệu đủ lớn và không bị mất cân bằng lớp.

**Phân tích mô hình:**

* **Decision Tree (i)**:Đây là một mô hình nhỏ bên trong mô hình tổng, chịu trách nhiêm .
* **Sample (i): Đây là kết quả của một cây quyết định i sau khi xử lý mô hình riêng biệt bên trong các thể mô hình đó**
* **Voting: là sự kiên diễn ra bên trong mô hình Random Forest dựa và các tính chất của dữ liệu và phản thiểu số của cây quyết định để đưa ra một chuẩn đoán xác thức, có căn cứ**
* **Test set**: một hay nhiều unseen data tức là dữ liệu chưa từng xử lý trong mô hình. Đây được coi như một bài kiểm tra không biết trước đề đầu tiên mà mô hình phải làm để từ đó có thể đánh giá chất lượng của mô hình. Đồng thời đưa ra nhưng phương ánh tối ưu để cải tiến nó
* **Training Set**: là tập hợp dữ liệu để đào tạo một mô hình một tệp test set sau khi qua thực nghiệm cũng có thể thể làm training set và tất nhiên nó không thể làm test set nữa. Nói đơn giản hơn Training set chính là tổng hợp các bài kiểm tra của năm trước và test set là bài kiểm tra cần phải làm để kết thúc môn, nếu bạn có nhiều training set thì test set sẽ đạt một kết quả tốt
* **Prediction:** đây làm chuẩn đoán mà mô hình đưa ra sau quá trình đào tạo

**Mô hình thường được áp dụng cho các loại bài toán như sau:**

* Các bài toán thuộc dạng [Classification](https://www.edureka.co/blog/classification-in-machine-learning/#:~:text=MNIST%20Digit%20Classification-,What%20is%20Classification%20In%20Machine%20Learning,as%20target%2C%20label%20or%20categories.) trên quy mô dữ liệu lớn, không gò bó về dữ liệu
* Mô hình cũng áp dụng các loại bài toán [Regression](https://www.investopedia.com/terms/r/regression.asp). Nhưng không được đánh giá cao

**Ưu điểm:**

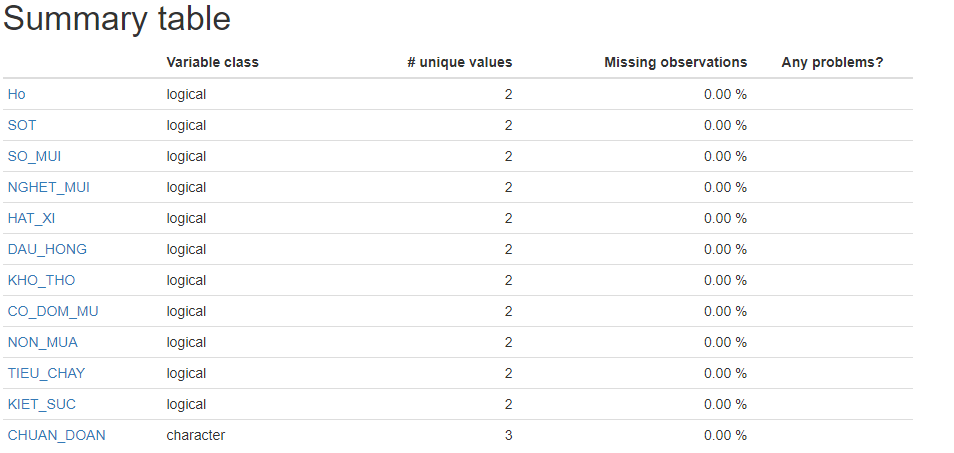
* Tỷ lệ chính xác cao dựa trên số dữ liệu
* Có thể tự động xử lý các dữ liệu bị khuyết
* Mô hình triển khai rõ ràng dễ dàng tinh chỉnh
* Có thể áp dụng Validation và Cross Validation để tránh tình trạng Overfitting

**Nhược điểm:**

* Cần một lượng dữ diệu đủ lớn để đào tạo mô hình
* Dữ liệu cần được làm sạch và được gắn nhãn
* Là một trong những thuật toán Blackbox nên Random Forest trở nên cực kỳ khó khăn trong việc diễn giải phương thức tạo ra kết quả
  1. **Tiến hành xây dựng mô hình:**
  2. **EDA**

Như đã nói ở trên, trước khi tiến hành xây dựng một hình chúng ta cần phải biết chúng ta đang có những gì từ đó có thể đút kết ra và xây dựng các đặc trưng cho các lớp cần phân loại trong dữ liệu. Vì trong bài này có đến 2 tập dữ liệu được sử dụng và hầu hết chúng đã được làm sạch (\*\*), và được gán nhán nên tôi sẽ đi sơ lược về tập dữ liệu ta sẽ làm việc.

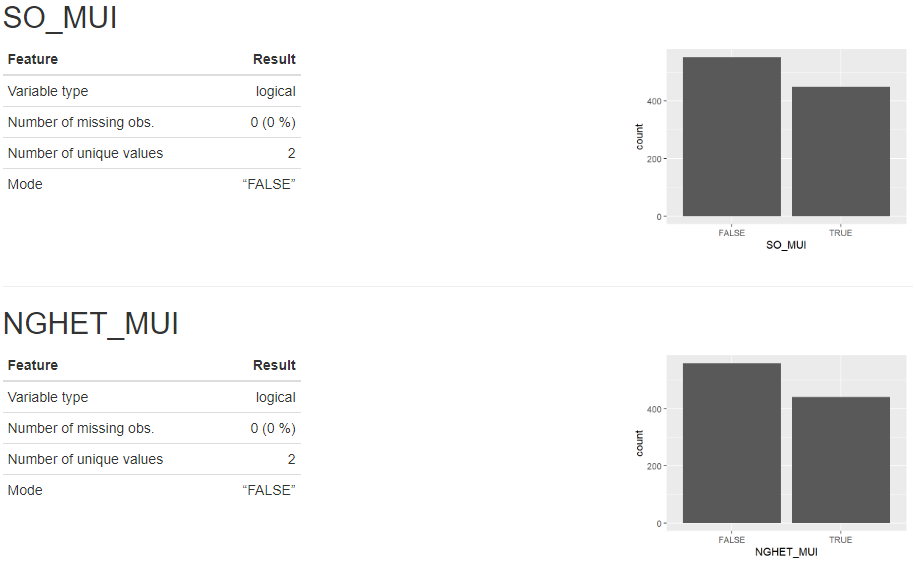
* SAR-COVID2: dùng thư viện dataMaid trong R tôi tạo một bản cáo cụ thể về dữ liệu mà tôi đang tiếp cận và lưu nó một file trên hệ thống

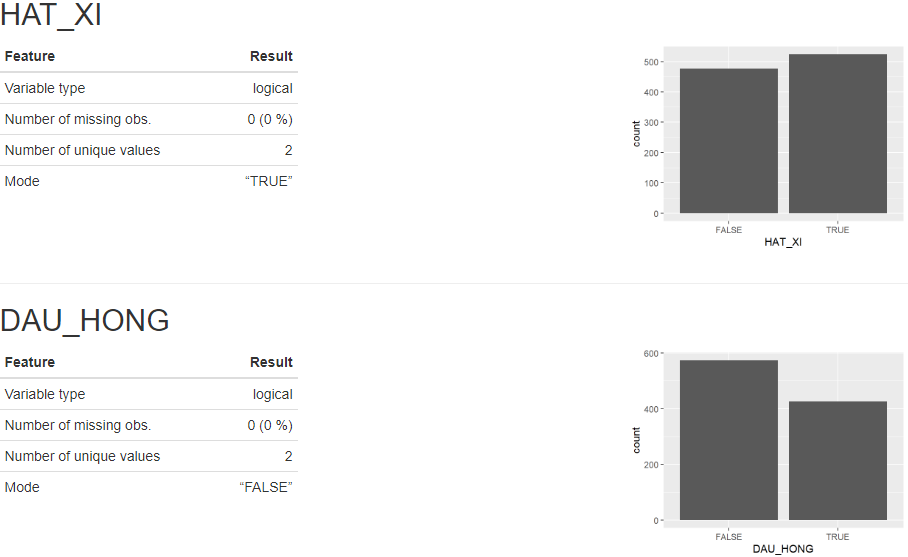


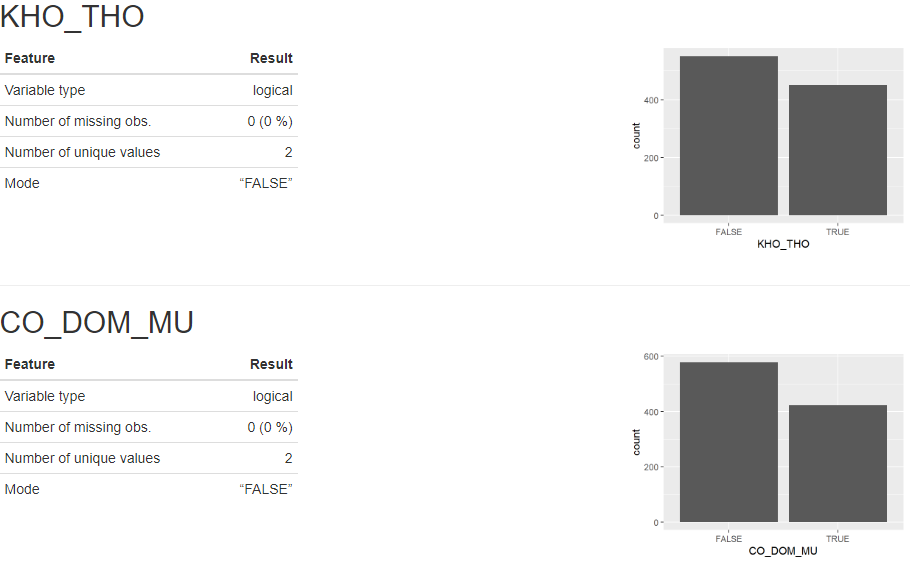
Có thể thấy dữ liệu được thể hiện đầy đủ thông qua từng loại kiểu dữ liệu , dữ liệu bị khuyết và vấn đề ở dữ liệu

Tiếp theo là một phần cho mỗi biến. Đây là phần dành cho chỉ số cụ thể, theo bảng tóm tắt ở trên có thể có vấn đề.

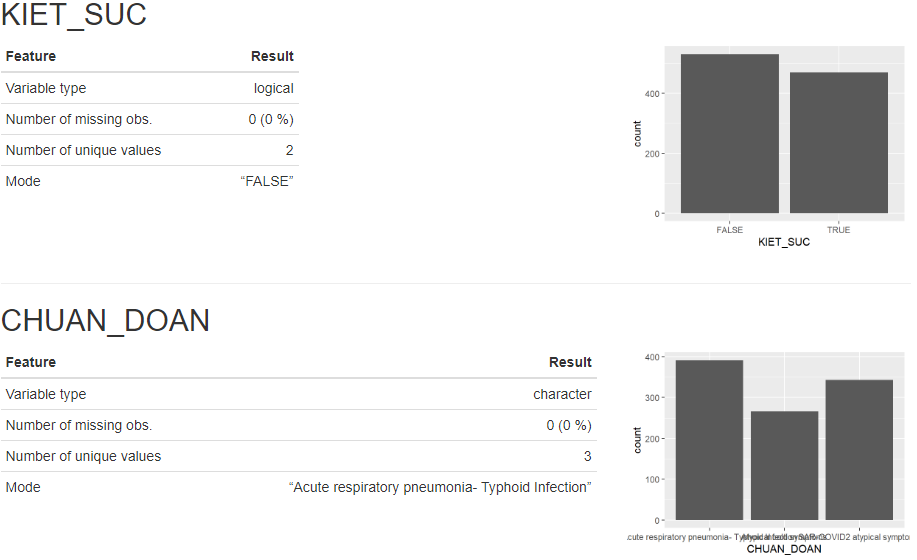








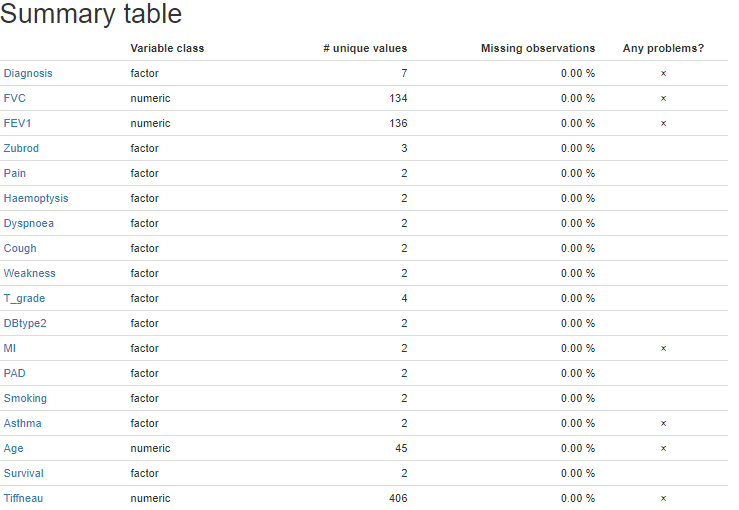


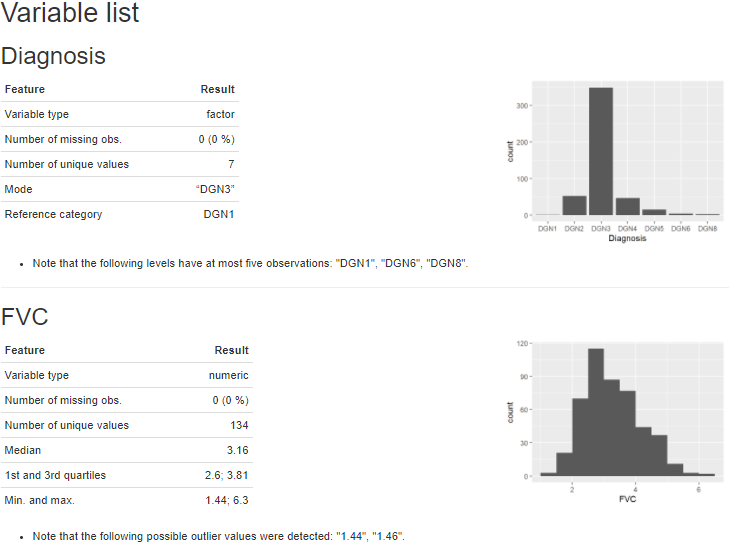


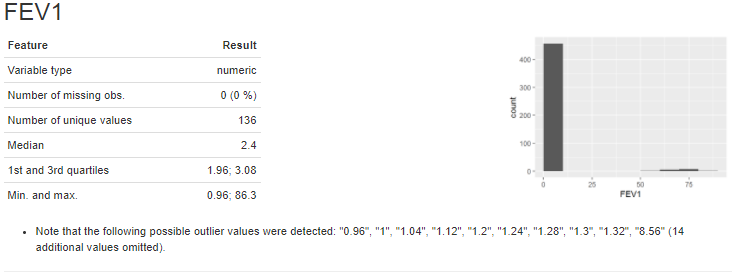
Có thể nhận ra rằng dự liệu đa số là kiểu dữ liệu logical chỉ có nhãn là kiểu ký tự và hơn nữa dữ liệu có xu hướng khá cân bằng không bị chênh lệch quá 15% cho mỗi kiểu, trừ dữ liệu nhãn CHUAN\_DOAN đây là dự liệu được kết luận dựa trên những dữ liệu được kê khai trước đó và một trong số dữ liệu này sẽ là outcome của mô hình. Vì vậy dữ liệu CHUAN\_DOAN có sự mất cân bằng lớp nhẹ được hiển thị trong biểu đồ

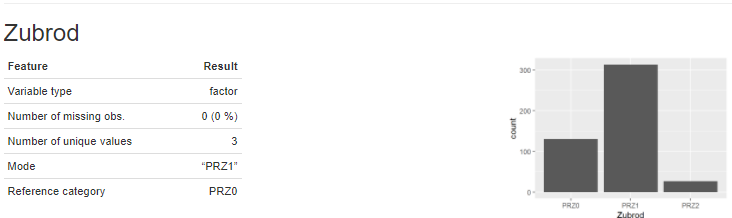
Như vậy bằng một cách thần kỳ nào đó thư viện này đã cho ta có một cái nhìn cực kỳ tổng quát thông qua các thống kê dưới đạng biểu đồ như vậy có thể thấy được

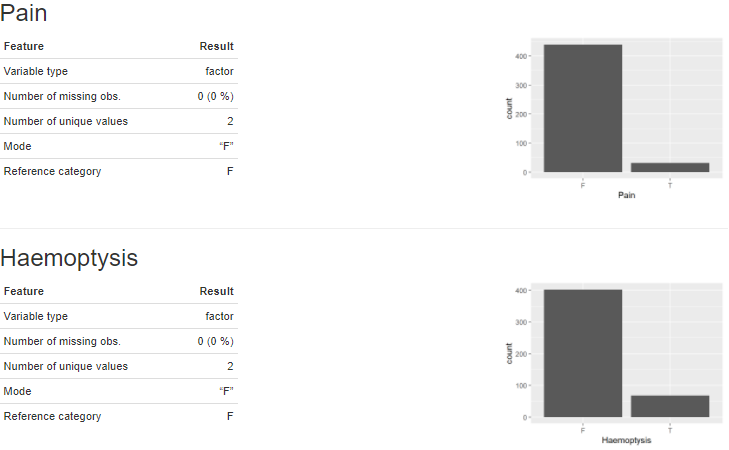
* DeadCovid: cũng như trên đây cũng là một tập dữ liệu phân loại và dự đoán kết quả và tương tự EDA của tập dữ liệu chắc chắn cũng tương tự như tập dữ liệu trên



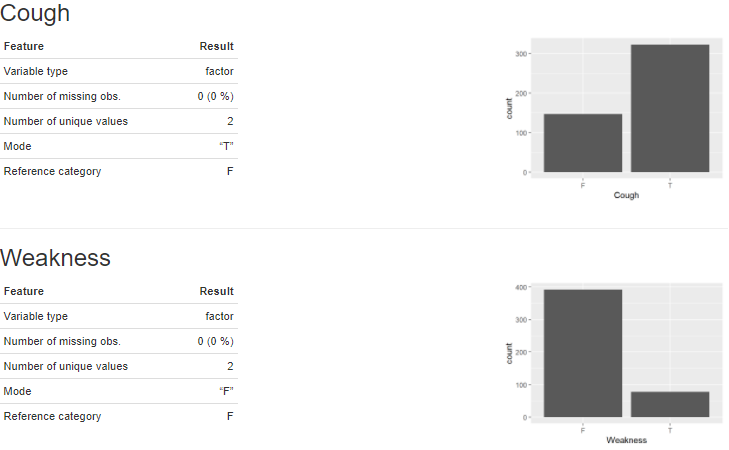
****

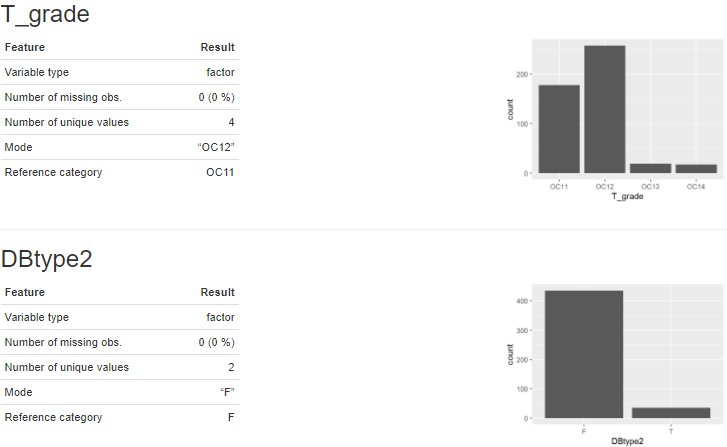
****

****

****

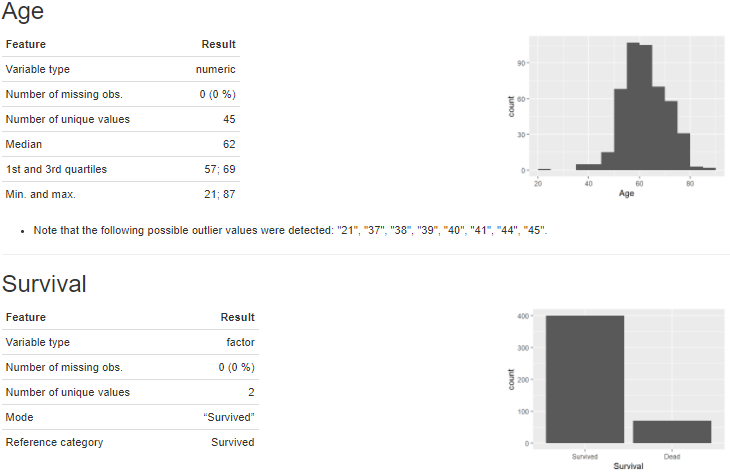
****

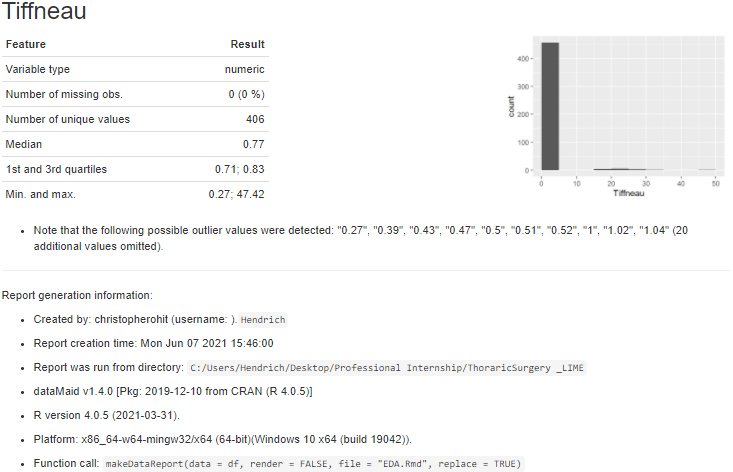
****

****

****

****

****

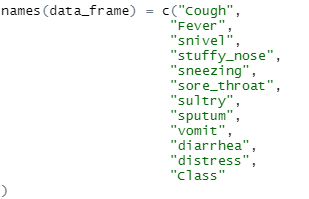


Nhìn vào sơ lược thống kê của dữ liệu, ta có thể xác định sự chênh lệch rõ ràng giữa các features và sự mất cân bằng khá lớn trong lớp phân loại. Cụ thể là lớp Survival, đây là lớp đưa ra quyết định cho việc tiên lượng sống sót ở các bệnh nhân nhưng lớp này bị mất cân bằng lớp nghiêm trọng có tới gần 410 kết quả sống sót nhưng vỏn vẹn chỉ có hơn 50 kết quả cho việc tử vong. Điều đáng nói ở đâu trong y học dữ liệu cho việc dương tính với các trường hợp chuyển biến lâm sàng thực sự rất quan trọng và cần thiết, nó là tiền đề cho việc đào tạo mô hình và chuẩn đoán. Hơn thế nữa nó cũng là mục đích của việc áp dụng Machine Learning vào trong y học

* 1. **Tiền xử lý:**

Trong giai đoạn này ta tiến hành các bước xử lý dữ liệu như là loại bỏ các feature không đặc trưng hay các feature gây nhiễu hoặc có thể gán tên cho các feature để tiện cho việc xử lý.

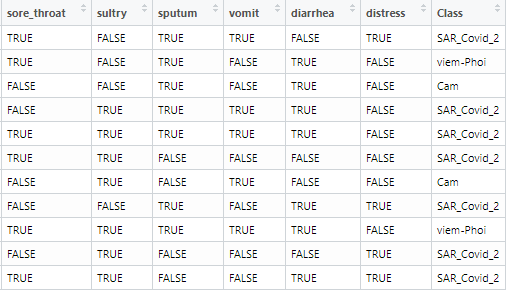
Trong các tập dữ liệu tôi xử lý trong bài báo cáo này hầu như không cần thiết để tối ưu hóa gì nữa chỉ cần đặt lại tên cho các feature và gán lại nhãn cho chúng để việc nắm bắt dữ liệu trở nên trực quan hơn. Riêng tập dữ liệu WuhanVirus có một vài đối tượng dữ liệu gây nhiễu và không đặc chưng nhưng chúng lại là đặc tính phổ biến của các lớp cần phân loại nên tôi quyết định không loại bỏ các feature này dù hậu quả có thể là việc mô hình sẽ bị giảm độ chính xác đáng kể.



|  |
| --- |
|  |

Tiến hành gán lại nhãn cho lớp phân loại:





Việc này tương tự cũng sẽ được lặp lại với tập dữ liệu khác. Như thể tôi đã hoàn thiện xong bước xử lý dữ liệu. Trong thực tế, việc xử lý dữ liệu rất phức tạp vì dữ liệu được đưa vào không sạch như chúng ta nghĩ. Lấy một ví dụ cơ bản: Trong thực tế, việc tự tay tạo ra những món ăn lẩu bò hấp dẫn người nhìn từ những nguyên liệu tươi sống sẽ phức tạp hơn rất nhiều so với việc đi vào quán ăn gia đình và bước ra với túi nước lẩu cùng với ít bún và mớ rau củ. Thật vậy những dữ liệu mà ta sử dụng hiện nay trong việc giảng dạy và thực hành đã được xử lý gọn gàn nhất nên bước xử lý dữ liệu này có thể trở nên không cần thiết. Dữ liệu của tôi cũng vậy đây là 2 tập dữ liệu tôi trích xuất từ chính Công ty tôi đang thực tập, nó đã được gắn nhãn và xử lý một cách hoàn chỉnh trước khi tới tay của tôi.

*Vậy câu hỏi cần được đặt nếu một lượng lớn tuồng vào và chúng ta không xử lý nó và tiến hành xây dựng mô hình trên dữ liệu đó có được không?* Câu trả lời là không. Chúng ta không thể cứ để những liệu tạp nham như thế mà xây dựng mô hình được, nếu như vậy thì công đoạn EDA thực sự là không cần thiết. EDA là công đoạn cho chúng ta một cái nhìn toàn cục về dữ liệu, nó gợi cho ta những vấn đề trong dữ liệu cần phải được xử lý. Và việc của ta là xử lý nó, chẳng hạn qua bước EDA ta thấy trong dữ liệu X features Y đang bị khuyết n dữ liệu, từ đó ta có thể chọn phương pháp để giải quyết vấn đề như loại bỏ dữ liệu khuyết hay thay thế nó bằng dữ liệu tiệm cận. Vậy ta đã có thể xử lý vấn đề của EDA. Hơn nữa việc gán nhãn dữ liệu đây là công việc tương đối khó khăn trong việc xây dựng mô hình. Đặc biệt là các dữ liệu mang nặng tính đặc trưng và tính phổ biến trong phân loại như Y học.

* 1. **Dựng mô hình Random Forest:**

1. **Mô hình chuẩn đoán bệnh**

Mục tiêu chính của nghiên cứu phát triển mô hình này là đánh giá ảnh hưởng và tính chính xác của thuật toán Rừng ngẫu nhiên được áp dụng cho một bài toán phân loại nhị phân. Nghiên cứu của tôi sẽ bao hàm bộ dữ liệu triệu chứng điển hình cho bệnh liên quan đặc biệt đến phổi và SAR-COVID2 (Bệnh viện Đại Học Y Dược TP.HCM). Bộ dữ liệu này được thu thập hồi cứu tại Đại học y dược TP.HCM cho những bệnh nhân đã trải qua giai đoạn nghiên cứu và chuẩn đoán bệnh thông qua các biểu hiện đặc trung không qua xét nghiệm của các bác sĩ lâm sàng trong giai đoạn 2020 đến 2021. Nhiệm vụ phân loại của chúng tôi nhằm đưa ra tỷ lệ mắc bệnh của những bệnh nhân đó, dựa trên các dấu hiệu điển hình của bệnh và tình trạng sức khỏe của họ.

Đầu tiên, tôi sẽ load tất cả các thư viện cần thiết, sau đó thiết lập version cho Java và config lại đường dẫn cho việc thiết lập môi trường làm việc.

|  |
| --- |
| library(tidyverse)  Sys.setenv(JAVA\_HOME="D:\\ZZZ Program Files\\Java")  library(rJava)  library(dataMaid)  library(explore)  library(keras)  library(dplyr)  require(foreign)  require(farff)  library(ggplot2)  library(tidyr)  library(tibble)  setwd("C:/Users/Hendrich/Desktop/Professional Internship/Predict\_SAR\_COVID2\_Base-on\_Sympton\_h2o") |

Bây giờ, tôi tải tập dữ liệu và thực hiện phân tích mô tả EDA, và bước này đã được thực hiện bên trên một cách chi tiết nên ở đây tôi sẽ thể hiện nó dưới dạng Terminal.

|  |
| --- |
| df <- read.csv("WuhanVirus.csv")  makeDataReport(df,  render = FALSE,  file = "EDA.Rmd",  replace = TRUE)  Hmisc::describe(df) |

|  |
| --- |
| df  12 Variables 1000 Observations  --------------------------------------------------------------------------------------------  Ho  n missing distinct  1000 0 2    Value FALSE TRUE  Frequency 513 487  Proportion 0.513 0.487  --------------------------------------------------------------------------------------------  SOT  n missing distinct  1000 0 2    Value FALSE TRUE  Frequency 568 432  Proportion 0.568 0.432  --------------------------------------------------------------------------------------------  SO\_MUI  n missing distinct  1000 0 2    Value FALSE TRUE  Frequency 551 449  Proportion 0.551 0.449  --------------------------------------------------------------------------------------------  NGHET\_MUI  n missing distinct  1000 0 2    Value FALSE TRUE  Frequency 559 441  Proportion 0.559 0.441  --------------------------------------------------------------------------------------------  HAT\_XI  n missing distinct  1000 0 2    Value FALSE TRUE  Frequency 476 524  Proportion 0.476 0.524  --------------------------------------------------------------------------------------------  DAU\_HONG  n missing distinct  1000 0 2    Value FALSE TRUE  Frequency 574 426  Proportion 0.574 0.426  --------------------------------------------------------------------------------------------  KHO\_THO  n missing distinct  1000 0 2    Value FALSE TRUE  Frequency 550 450  Proportion 0.55 0.45  --------------------------------------------------------------------------------------------  CO\_DOM\_MU  n missing distinct  1000 0 2    Value FALSE TRUE  Frequency 578 422  Proportion 0.578 0.422  --------------------------------------------------------------------------------------------  NON\_MUA  n missing distinct  1000 0 2    Value FALSE TRUE  Frequency 574 426  Proportion 0.574 0.426  --------------------------------------------------------------------------------------------  TIEU\_CHAY  n missing distinct  1000 0 2    Value FALSE TRUE  Frequency 502 498  Proportion 0.502 0.498  --------------------------------------------------------------------------------------------  KIET\_SUC  n missing distinct  1000 0 2    Value FALSE TRUE  Frequency 530 470  Proportion 0.53 0.47  --------------------------------------------------------------------------------------------  CHUAN\_DOAN  n missing distinct  1000 0 3  Acute respiratory pneumonia- Typhoid Infection (356, 0.356), Atypical cold symptoms (323,  0.323), SAR-COVID2 atypical symptoms (321, 0.321)  --------------------------------------------------------------------------------------------  > df <- read.csv("WuhanVirus.csv")  > Hmisc::describe(df)  df  12 Variables 1000 Observations  --------------------------------------------------------------------------------------------  Ho  n missing distinct  1000 0 2    Value FALSE TRUE  Frequency 513 487  Proportion 0.513 0.487  --------------------------------------------------------------------------------------------  SOT  n missing distinct  1000 0 2    Value FALSE TRUE  Frequency 568 432  Proportion 0.568 0.432  --------------------------------------------------------------------------------------------  SO\_MUI  n missing distinct  1000 0 2    Value FALSE TRUE  Frequency 551 449  Proportion 0.551 0.449  --------------------------------------------------------------------------------------------  NGHET\_MUI  n missing distinct  1000 0 2    Value FALSE TRUE  Frequency 559 441  Proportion 0.559 0.441  --------------------------------------------------------------------------------------------  HAT\_XI  n missing distinct  1000 0 2    Value FALSE TRUE  Frequency 476 524  Proportion 0.476 0.524  --------------------------------------------------------------------------------------------  DAU\_HONG  n missing distinct  1000 0 2    Value FALSE TRUE  Frequency 574 426  Proportion 0.574 0.426  --------------------------------------------------------------------------------------------  KHO\_THO  n missing distinct  1000 0 2    Value FALSE TRUE  Frequency 550 450  Proportion 0.55 0.45  --------------------------------------------------------------------------------------------  CO\_DOM\_MU  n missing distinct  1000 0 2    Value FALSE TRUE  Frequency 578 422  Proportion 0.578 0.422  --------------------------------------------------------------------------------------------  NON\_MUA  n missing distinct  1000 0 2    Value FALSE TRUE  Frequency 574 426  Proportion 0.574 0.426  --------------------------------------------------------------------------------------------  TIEU\_CHAY  n missing distinct  1000 0 2    Value FALSE TRUE  Frequency 502 498  Proportion 0.502 0.498  --------------------------------------------------------------------------------------------  KIET\_SUC  n missing distinct  1000 0 2    Value FALSE TRUE  Frequency 530 470  Proportion 0.53 0.47  --------------------------------------------------------------------------------------------  CHUAN\_DOAN  n missing distinct  1000 0 3  Acute respiratory pneumonia- Typhoid Infection (352, 0.352), Atypical cold symptoms (327,  0.327), SAR-COVID2 atypical symptoms (321, 0.321)  -------------------------------------------------------------------------------------------- |

Như đã thấy các yếu tố quyết định đến việc dự đoán gồm 12 biến:

Tên biến được tạo bằng tiếng việt không dấu nên không cần nói chi tiết, ta chỉ chú ý về biến chuẩn đoán đây là biến nhãn của dữ liệu. Tại đây ta thấy có sự mất cân lớp khá khá. Lớp Acute respiratory pneumonia-Typhoid Infection (Viêm phổi triệu chứng không điểu hình) và Atypical cold symptoms (cảm cúm dấu hiệu phổ biến) sẽ được quy vào mặt ổn định, nghĩa là bệnh nhân được chuẩn đoán mắc phải những bệnh này sẽ được xử lý với một phương pháp phổ biến tương đối như nhau. Nhưng đối với SAR-COVID2 atypical symptoms thì lại là một trường hợp đặc biệt cần chú ý. Đây là đối tượng của bài nghiên cứu, là một loại bệnh mới xuất hiện với dấu hiệu khó phân biệt với các bệnh liên quan đến phổi nên mục đích chính của nghiên cứu là phân loại bệnh nhân bị nhiễm COVID và công việc quan sát của tôi sẽ chỉ xoay quanh có định về đối tượng này. Đã được mô tả ở EDA nếu ta đồng quy 2 đối tượng ARP-TI và ACS là một thì tỷ lệ trong dữ liệu giữa nó và phần còn lại là gần với và đây là một tỷ lệ được đánh giá là không quá mất cân bằng.   
Lưu ý: đây là dữ liệu được lấy từ rút ra từ nghiên cứu thực và dựa vào thực thế để chứng minh với bệnh SAR-COVID2 là một bệnh truyền nhiễm có tính lây lan cao và nhanh thế nên việc tỷ lệ chênh lệch như vầy được gọi là hợp lý.

Tiếp theo cho việc khởi tạo mô hình là khởi tạo mô hình RandomForest trong R và thư viện “caret” sẽ được sử dụng để phân tách dữ liệu, đây sẽ thư viện ổn định nhất trong việc đảm bảo tỷ lệ các lớp giống nhau trong tập hợp trainging và testing. Với việc giữ nguyên nhãn của đối tượng (không góp trong bất cứ tính năng phân loại nào ở tập test) để thuận tiện cho việc đối chiếu và so sánh, tôi hy vọng dữ liệu không cân bằng sẽ không ảnh hưởng đến việc xác định cũng như dự đoán của mô hình.

|  |
| --- |
| library(caret)  set.seed(1234)  idtest = caret::createDataPartition(y = data\_frame$Class,p = 800/1000,list = FALSE)  trainset = data\_frame[idtest,]  testset = data\_frame[-idtest,] |

Mô hình được tạo ra khá nhanh chóng. Khi kiểm định mô hình trên chính tập Training set độ chính xác của mô hình tương đối ổn, tạm chấp nhận được với tỷ lệ sai 37,1% cho bệnh COVID, 48,8% cho viêm phổi và 65,25% cho bệnh cảm.

|  |
| --- |
| library(randomForest)  trainset$Class = factor(trainset$Class)  rfmod = randomForest(Class~.,data=trainset,ntree=700,  mtry = sqrt(11),  replace=TRUE,  localImp=TRUE  )  rfmod |

|  |
| --- |
| Call:  randomForest(formula = Class ~ ., data = trainset, ntree = 700, mtry = sqrt(11), replace = TRUE, localImp = TRUE)  Type of random forest: classification  Number of trees: 700  No. of variables tried at each split: 3  OOB estimate of error rate: 49.19%  Confusion matrix:  SAR\_Covid\_2 viem-Phoi Cam class.error  SAR\_Covid\_2 173 93 9 0.3709091  viem-Phoi 102 160 51 0.4888179  Cam 37 102 74 0.6525822 |

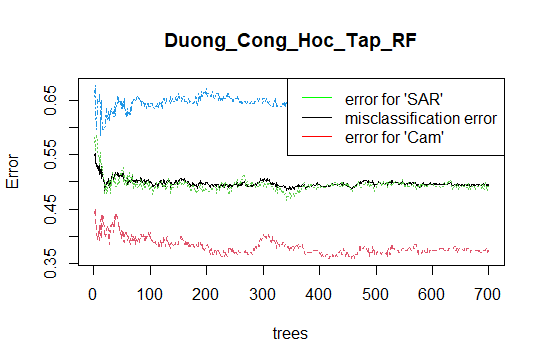
Khi kiểm tra trên tập testset độc lập thì mô hình đạt độ chính xác trong khoảng ổn định với BAC lần lượt cho mỗi thành phần là 71.88%, 59.57% và 60.38%. Để lý giải cho việc tỷ lệ dự đoán được thấy như thế này có thể quy cho việc phân lớp nghĩa là đứng trước một ngã ba đường bạn sẽ ra quyết định nhanh chóng, dễ dàng và có tỷ lệ chính xác hơn là một ngã tư. Dưới góc nhìn của một nghiên cứu sinh về Data science tôi cho rằng trong y học nói riêng và trong mọi lĩnh vực thực tiễn nói chung, ta chỉ nên xác định một và chỉ tâm điểm duy nhất và đó cũng là thứ ta cần nhắm đến mọi thứ khác nếu có liên quan đặc biệt chặt chẽ đến thứ ta nghiên cứu thì nó thuộc về Dương tình hoặc không là âm tính. Như vậy trong một hồi lâu suy nghĩ về những phương pháp gia tăng độ chính xác của một mô hình tôi đã nhận ra mình đã đi quá lòng vòng và xử lý quá nhiều thứ không cần thiết nên có thể nói giải pháp trên là thứ tôi đúc kết được trong lần nghiên cứu thực tập này, nhưng thay vì thay đổi các tham số tham gia dữ liệu tôi vẫn tiến hành để nguyên vì nó vẫn còn nhiều thứ hay ho trong việc nghiên cứu và phân tích.

|  |
| --- |
| testpred = predict(rfmod, testset)  testset$Class = as.factor(testset$Class)  confusionMatrix(factor(testpred),  reference = testset$Class,  # positive = "SAR\_Covid\_2",  mode = "everything") |

|  |
| --- |
| Confusion Matrix and Statistics  Reference  Prediction SAR\_Covid\_2 viem-Phoi Cam  SAR\_Covid\_2 50 26 13  viem-Phoi 17 42 25  Cam 1 10 15  Overall Statistics    Accuracy : 0.5377  95% CI : (0.4658, 0.6084)  No Information Rate : 0.392  P-Value [Acc > NIR] : 2.216e-05    Kappa : 0.2854    Mcnemar's Test P-Value : 0.000331  Statistics by Class:  Class: SAR\_Covid\_2 Class: viem-Phoi Class: Cam  Sensitivity 0.7353 0.5385 0.28302  Specificity 0.7023 0.6529 0.92466  Pos Pred Value 0.5618 0.5000 0.57692  Neg Pred Value 0.8364 0.6870 0.78035  Precision 0.5618 0.5000 0.57692  Recall 0.7353 0.5385 0.28302  F1 0.6369 0.5185 0.37975  Prevalence 0.3417 0.3920 0.26633  Detection Rate 0.2513 0.2111 0.07538  Detection Prevalence 0.4472 0.4221 0.13065  Balanced Accuracy 0.7188 0.5957 0.60384 |

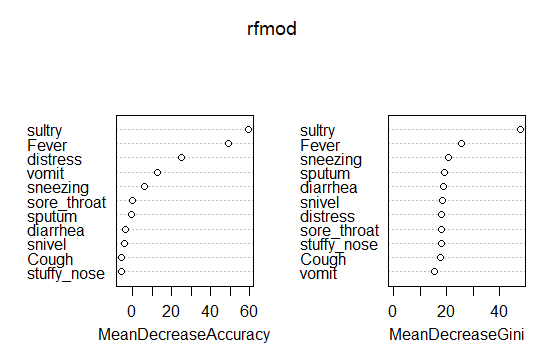
Và đây là biểu đồ giao động của các cây trong quá trình đào tạo mô hình khi tăng kích thước từ 0 đến 700 cây và ở mức 500 ta thấy có sự ổn định nhẹ trong việc phân lớp cho SAR và phân lớp ở bệnh “Cảm” có dấu hiệu nhiễu tương đối. Để lý giải cho hiện tượng này thì có thể tôi có thể đưa ra một số vấn đề mang nặng đặc tính của y học như sau. SAR COVID2 như mọi người đã biết thì đây một bệnh liên quan một đặc biệt tới đường hô hấp và cụ thể là gây tổn thương phổi một cách nghiêm trọng, được sinh sau để muộn nên COVID được kế thừa hầu hết các đặc tính của người tiền nhiệm trước đó là SARS (servere acute respiratory syndrome) một loại bệnh gây tỷ lệ tử vong khá cao cũng có xuất xứ từ Trung Quốc nhưng là ở Quảng Đông. Vốn là một bệnh liên qua trực tiếp đến hô hấp và trao đổi khí giữa các phế nang, SARS giết chết con mồi theo nguyên lý và các thức tấn công của HIV/AIDS với phổi của họ dẫn tới việc bệnh nhân suy hô hấp cấp trong trường hợp xấu nhất dẫn đến tử vong. Như thế có thể thấy được người kế nhiệm COVID2 là một trường hợp đặc biệt nguy hiểm, thông qua những con số không biết nói dối hơn 3,771,250 người tử vong ta có thể thấy được một cuộc thanh trừng thật sự đang xảy chỉ sau chiến tranh. Quay lại chủ đề chính vốn là một bệnh hô hấp nên nó thừa hưởng khá nhiều tính trạng trội về triệu chứng phổ biến của một bệnh hô hấp nên việc gây nhiễu giữa một bệnh về hô hấp biến và một bệnh phổi hiếm gặp là hoàn toàn có cơ sở để giải thích . Ngoài lề một chút, có một câu hỏi trong quá trình nghiên cứu cũng như đào tạo mô hình tôi đã gặp và cảm thấy khá khá hứng thú với nó “*Có một vài trường hợp giữa 2 bệnh nhân, tất cả triệu chứng giữa 2 họ đều giống nhau nhưng sau khi có kết quả xét nghiệm một trong số họ lại dương tính người còn lại là âm tính. Tại sao vậy ?*” câu trả lời tôi sẽ giải đáp trong phần kết luận của mô hình

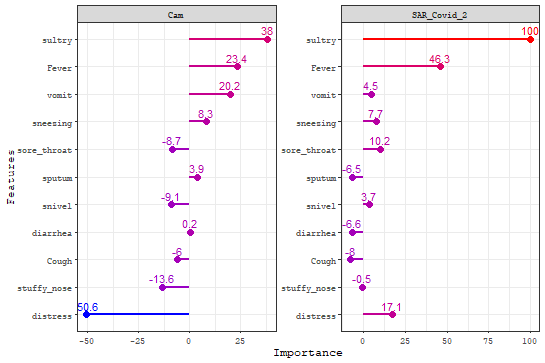
|  |
| --- |
| plot(rfmod,  main = "Duong\_Cong\_Hoc\_Tap\_RF")  legend("topright",  c("error for 'SAR'",  "misclassification error",  "error for 'Cam'"),  lty = c(1,1,1),  col = c("green" , "black" , "red"))  varImpPlot(rfmod) |



Tiếp theo ta tiến hành khảo sát vai trò của biến trong mô hình, thư viện caret sẽ giúp ta làm điều đó

|  |
| --- |
| varImpPlot(rfmod)  varImp(rfmod)%>%as.data.frame()%>%  mutate(Feature=rownames(.))%>%  gather(Cam,SAR\_Covid\_2,key="Label",value="Importance")%>%  mutate(Importance=100\*Importance/max(.$Importance))%>%  ggplot(aes(x=reorder(Feature,Importance),  y=Importance,  fill=Importance,  color=Importance))+  geom\_segment(aes(x=reorder(Feature,Importance),  xend=Feature,  y=0,  yend=Importance),  size=1,  show.legend = F)+  geom\_point(size=2,show.legend = F)+  scale\_x\_discrete("Features")+  coord\_flip()+  scale\_fill\_gradient(low="blue",high="red")+  scale\_color\_gradient(low="blue",high="red")+  geom\_text(aes(label = round(Importance,1)),  vjust=-0.5,size=3)+  facet\_wrap(~Label,ncol=2,scales="free")+  theme\_bw(base\_family = "mono",8) |





Một số biến tốt (có vai trò quan trọng, đóng góp đáng kể vào việc phân lọai) sẽ có tần suất hiện diện cao hơn trong mô hình (xuất hiện trong những cây tốt, tương ứng với nhiều nodes). Dựa vào biểu đồ trên có thể thấy được rằng biến Sultry là quan trọng nhất còn biến distress ít tác dụng nhất. Dựa vào đây có thể diễn giải phần nào về cơ chế hoạt động của một mô hình. Nhưng việc diễn giải mô hình không phải dừng ở đây với library randomForest Explainer ta có thể tiến hành đào sâu hơn về vai trò của từ features bên trong mô hình random Forest. Đầu tiến là kết quả định danh nhanh, tôi lọc ra danh sách K biến quan trọng nhất đã tham gia vào mô hình

|  |
| --- |
| rfmod %>%  important\_variables(k = 7, ties\_action = "draw") |

|  |
| --- |
| [1] "diarrhea" "distress" "Fever" "sneezing" "sputum" "sultry"  [7] "sore\_throat |

Thí dụ, k=7 tương ứng 7 biến quan trọng nhất, theo thứ tự từ Trái sang phải. Kết quả này có thể rất quan trọng, nếu bạn có trong tay đến hàng ngàn, thậm chí hàng trăm ngàn biến chứ không phải chỉ có 11. Như vậy, Mô hình Random Forest cũng là một giải pháp nhằm chọn lọc biến số cho bài toán phân loại, bất kể algorithm mà bạn muốn áp dụng sau đó là gì. Với tốc độ nhanh thế này, bạn hoàn toàn có thể dùng RF để thăm dò dữ liệu và chọn lọc biến số.

Hơn thế nữa, package random Forest Explainer còn cung cấp cho chúng ta đến 6 chỉ số thống kê để đo lường vai trò quan trọng của từng biến, như sau:

Minimal depth: Chiều sâu của cây quyết định là kích thước của con đường dài nhất từ rễ cây (biến số đầu tiên) đến lá cây (một kết quả phân loại, sau nhiều lần phân nhánh). Tree depth thể hiện mức độ phức tạp của cấu trúc mô hình. Mỗi biến Xj nếu hiện diện trong một cây, sẽ có một giá trị minimal depth xác định, là một số nguyên dương (thí dụ từ 0 đến 11). Nếu Xj không hiện diện, thì giá trị này được tính là missing value (NA). Như vậy trong mô hình RF với 700 cây, phân bố của giá trị Min\_depth cho mỗi biến số có thể cung cấp thông tin về : tần suất hiện diện của biến số đó, sự phức tạp của quy luật mà biến đó tham gia (min\_depth cao). Đây là phân bố của một số nguyên dương, và ta có thể tính giá trị trung bình (mean of min\_depth) cho từng biến.

Lưu ý : Khi chọn chế độ lấy mẫu là « relevant trees », giá trị trung bình chỉ tính dựa trên số lượng cây mà biến đó có tham gia, không kể các missing value.

Accuracy decrease: của một biến Xj đo lường sự sụt giảm độ chính xác của mô hình cây khi Xj bị hoán chuyển. Một cách gián tiếp, chỉ số này cho thấy mức độ quan trọng của biến Xj trong mô hình.

Gini decrease: của biến Xj đo lường sự suy giảm trung bình của [Gini impurity](file:///C:\Users\Hendrich\Desktop\Professional%20Internship\Doc\Category\Gini.docx) tại mỗi node tương ứng với những lần phân chia dựa vào biến Xj này. Có thể hình dung là Gini decrease cho thấy ảnh hưởng quan trọng của Xj đối với quy luật phân nhóm dữ liệu, làm giảm xác suất phạm sai lầm khi dán nhãn cho các trường hợp.

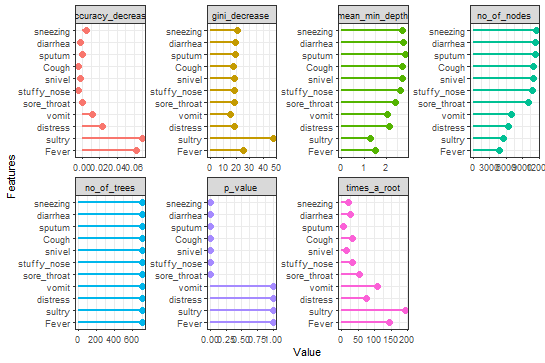
No\_of\_trees : đơn giản là tổng số cây có hiện diện biến Xj. Sự hiện diện của Xj trong nhiều cây cho thấy Xj là một tiêu chí quan trọng.

No of nodes : tổng số node (nút phân nhánh) có liên quan đến Xj trong mô hình Random Forest, ý nghĩa của chỉ số này tương tự như no\_of\_trees.

Time\_a\_root : tổng số cây mà trong đó biến Xj được sử dụng ngay từ rễ (lần phân nhánh đầu tiên trong mô hình cây dựa vào Xj).

P\_value : trị số p (1 bên) của kiểm định binomial dựa trên phân phối Binomial (tổng số nodes, P(số nodes do phân chia Xj)). Giả thuyết H0 của kiểm định này cho rằng việc Xj tham gia phân nhánh chỉ là do ngẫu nhiên. Đặt ngưỡng ý nghĩa cho p=0.05, ta có thể bác bỏ giả thuyết H0 này và tin rằng Xj thực sự có vai trò quan trọng.

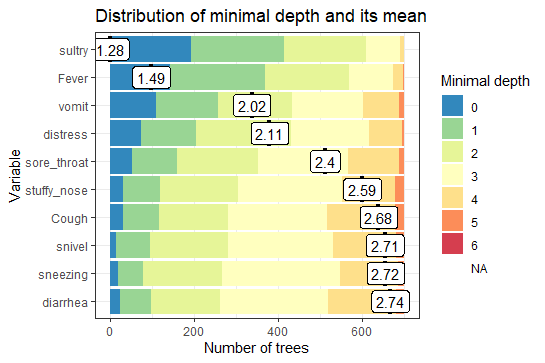
Từ dataframe chứa những chỉ số kể trên, ta có thể trình bày kết quả dưới nhiều hình thức biểu đồ. Đầu tiên là bar graph cho giá trị trung bình.



Riêng chỉ số Minimal depth distribution có thể được trình bày bằng hình vẽ sau:

|  |
| --- |
| mindepthdf = rfmod %>% min\_depth\_distribution() %>% as\_tibble()  mindepthdf %>% ggplot(aes(x = minimal\_depth,  fill = variable,  col= variable)) +  geom\_histogram(alpha=0.5) +  facet\_wrap(~variable, ncol = 3, scales = "free") +  my\_theme()  #  rlang::last\_error()  rfmod %>% min\_depth\_distribution() %>%  plot\_min\_depth\_distribution(mean\_sample = "all\_trees") +  scale\_fill\_brewer(palette = "Spectral", direction = -1) |

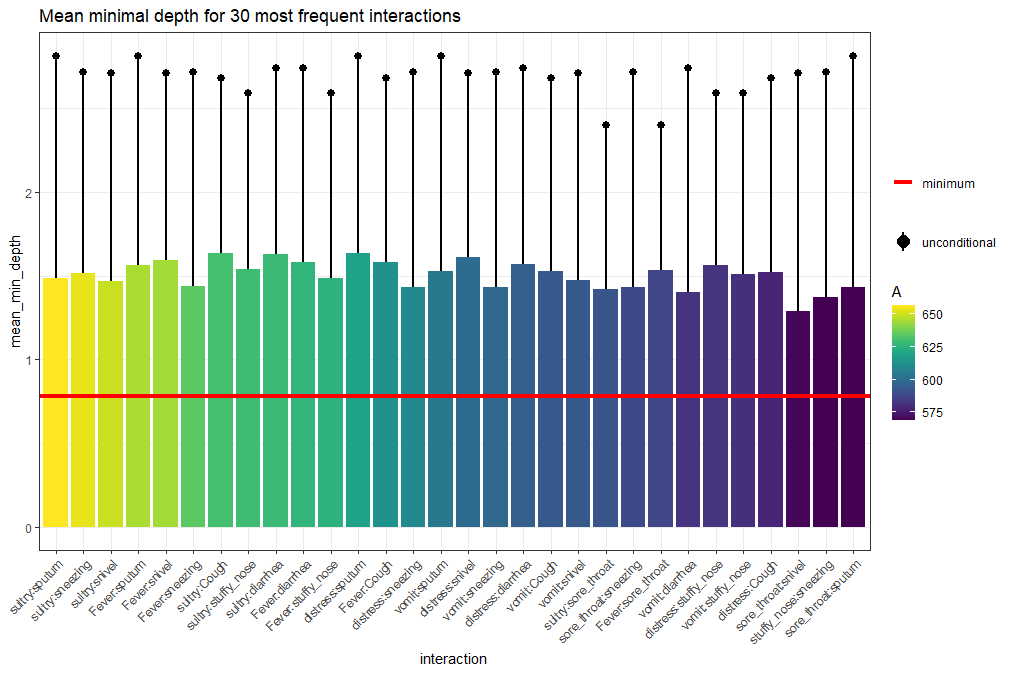




Nói lại một chút về Minimal depth of tree, như để giải nghĩa ở phía trên một biến có giá trị minimal depth trung bình cao nghĩa là xét về logic thì việc xử lý các biến này càng phức tạp và có thể tới trên 80% đây là một biến có độ phân bố phổ biến và gây nhiễu dữ liệu, và “diarhea” là một biến minh họa cụ thể.

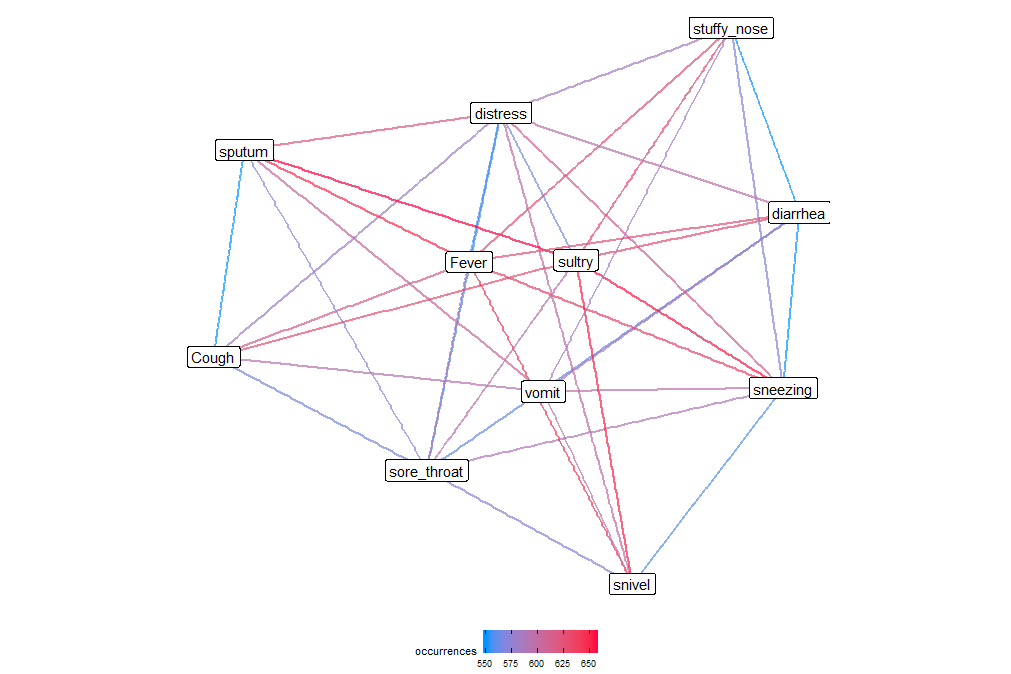
Không chỉ dừng lại ở mức độ đo lường vai trò của các biến riêng lẻ thư viện này còn cho phép ta khảo sát về sự tương tác giữa 2 biến. Sự tương tác xảy ra khi 2 biến cùng hiện diện trong một mô hình cây và tham gia vào sự phân nhánh ở các nodes.

|  |
| --- |
| mindprint = min\_depth\_interactions(rfmod, mean\_sample = "relevant\_trees", uncond\_mean\_sample = "relevant\_trees")  library(viridis)  mindprint %>%  plot\_min\_depth\_interactions() +  scale\_fill\_viridis("A") |



Khi quan sát nội dung của dữ liệu, kết quả phân tích tương tác biến, ta có thể nhận thấy nó có dạn một Network, tức là ta còn có thể trình bày các biến dưới dạng mạng lưới. Để thực hiện đầu tiên ta lọc những cặp tương tác nào có tần suất xuất hiện nhỏ hơn 550. Sau đó ta chuyển toàn bộ dự liệu khung thành một network, sau cùng dùng thư viện ggraph để mã hóa màu sắc cho các liên kết theo tần suất hiện diện ta được network như sau:

|  |
| --- |
| head(mindprint)  library(igraph)  library(ggraph)  gdf = filter(mindprint,  occurrences >= 550)  graph <- graph\_from\_data\_frame(gdf[,c(1:3,4,6)])  ggraph(graph,  layout = 'kk',  circular=F)+  geom\_edge\_link(aes(colour = occurrences),  show.legend = T,width=1,alpha=0.7)+  geom\_node\_label(aes(label = name))+  coord\_fixed()+  scale\_edge\_color\_gradient(high="#ff003f",low="#0094ff")+  scale\_edge\_width\_continuous()+  theme\_bare() |



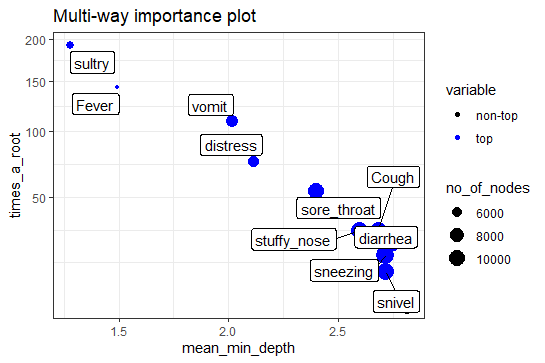
Network này cho chúng ta biết 2 thông tin cực kỳ quan trọng:

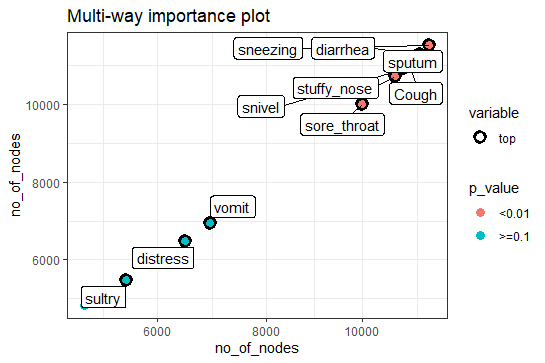
* Mức độ tương tác của các biến trong mô hình: những biến có liên kết tương tác chặt chẽ sẽ gom lại gần nhau thành một cụm và liên kết có màu đỏ, biến “sultry” nằm ở khu vực trung tâm được xem là quan trọng nhất, biến “sore\_throat” mặc dù có tới 7 liên kết nhưng lại là liên kết kém, đáng lý biến này chỉ có duy nhất 3 liên kết nhưng vì một lý do khá đặc biệt cũng có thể đặc trưng trong y học nên xuất hiện tình trạng này và lý dó cũng liên quan mật thiết đến câu hỏi được đặt ra trong báo cáo và sẽ được giải đáp trong phần đánh giá kết luận. Nhắc lại, một trong những lợi thế của mô hình quyết định đó là nó cho phép mô hình hóa những quan hệ tương tác phức tạp giữa nhiều biến, thậm chí vượt xa cả tầm hiểu biết của người dựng mô hình.
* Khả năng lọc bỏ tự động những biến không có vai trò quan trọng đối với mục tiêu mà ta cần nhắm đến. Do đó, khi 2 hay 3 biến đồng thời xuất hiện lặp đi lặp lại trong mô hình nhiều mô hình cây cho thấy một dấu hiệu rằng hoặc là chúng quan trọng hoặc là chúng có liên hệ với nhau. Tất nhiên khi chúng quan trọng chúng sẽ có một liên kết khá dày nhưng liên kết dày chưa chắc chúng quan trọng.

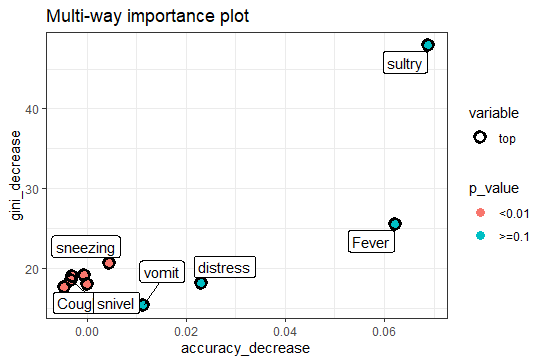
Như vậy, ta có thể áp dụng hình thức này như một giải pháp thay thế cho correlation matrix trong một mô hình tiên lượng hồi quy và phân loại, biểu đồ Network trên đây đưa ra bức tranh toàn cảnh về liên hệ giữa các biến số quan trọng nhất cùng hướng tới mục tiêu xác định.

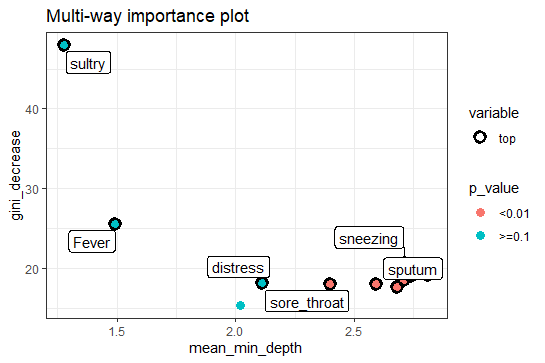
Sau đây ta tiến hành trình bày dữ liệu dưới dạng Multiway importance plot, biều đồ trình bày nhiều thông tin một lúc dưới dạng 2 chiều: tương quan giữa 2 chỉ số bất kỳ, phân phối chúng và giá trị p\_value của binomial test. Đây cùng với biểu đồ Network là nguyên nhân chính cho việc không sửa lại nhãn cho tập dữ liệu ban đầu bởi chỉ có như vậy ta mới thấy được sự phân bố, liên kết tương tác cực kỳ phức tạp của dữ liệu thực tế cần xử lý.

|  |
| --- |
| #1  plot\_multi\_way\_importance(rfmod,  x\_measure = "mean\_min\_depth",  y\_measure = "times\_a\_root",  size\_measure = "no\_of\_nodes",  min\_no\_of\_trees = 100)  #2  plot\_multi\_way\_importance(vimdf,  x\_measure = "no\_of\_nodes",  y\_measure = "no\_of\_nodes",  size\_measure = "p\_value",  min\_no\_of\_trees = 100)  #3  plot\_multi\_way\_importance(vimdf,  x\_measure = "accuracy\_decrease",  y\_measure = "gini\_decrease",  size\_measure = "p\_value",  min\_no\_of\_trees = 100)  #4  plot\_multi\_way\_importance(vimdf,  x\_measure = "mean\_min\_depth",  y\_measure = "gini\_decrease",  size\_measure = "p\_value",  min\_no\_of\_trees = 100) |









Từ những thống kê từ khi mở đầu cho đến hiện tại ta đã đủ những lý luận cho các bước hoạt động của mô hình từ đây mô hình có thể được áp dụng và diễn giải trước trước các bác sĩ và bên bệnh nhân một cách rõ ràng. Quay trở lại mô hình để sơ lược một chút ta hoàn toàn có thể nhìn nhận được biến “sultry” là cốt yếu góp phần tạo nên tính chính xác của mô hình và kiểm tra tính xác thực của nó, ta có thể nghiên cứu khoa hô hấp của y học để đưa ra kiểm chứng rằng triệu chứng của một dạng bệnh nhiễm trùng phổi là suy hô hấp, nhưng nó hoàn toàn không có nghĩa rằng bệnh nhân nhiễm trùng phổi đều suy hô hấp trong mọi giai đoạn của bệnh. Đây cũng là một trong những nguyên nhân thứ yếu gây nhiễu dữ liệu trong việc chuẩn đoán.

Có thể nói qua những gì thể hiện bên trên, ta đã bước đầu thành công trong việc tiến hành xây dựng một hình học máy cho dữ liệu thật và diễn giải chúng. Tuy còn rất nhiều khuyết điểm trong việc xây dựng mô hình (thứ mà tôi sẽ nêu rõ trong phần Đánh giá cuối báo cáo) nhưng đó vẫn là một nền móng khá vững chắc cho việc xây dựng ý tương trong Đồ án tốt nghiệp.

1. **Tiên Lượng tử vong**

Như vậy qua việc tiến hành thực nghiệm và quan sát ta đang có một số cái nhìn tổng quát về dữ liệu nhưng ngẫm lại những gì vừa làm, tình trạng chuẩn đoán cho căn bệnh vừa rồi tuy gặp rất nhiều vấn đề khó khăn mà chủ yếu quay quanh việc nhiễu dữ liệu còn lại thì với khả năng lây lan mạnh mẽ nó, bệnh đã quá phổ biến không còn hiếm nhưng nếu trong trường hợp tỷ lệ mắc bệnh với xác xuất 10% trên tổng dân số thì sẽ nhưng thế nào nhỉ? Một ví dụ không thể thực tế và liên kết hơn khi tôi tiến hành tiên lượng tử vong cho bệnh nhân trong quá trình điều trị bệnh COVID2. Tại sao à ? Vì nếu xét trên tập bệnh nhân bị nhiễm bệnh là 175,216,127 và người tử vong là 3,778,311 (theo số liệu thực tế ngày 10/06/2021 tại trang [này](https://www.worldometers.info/coronavirus/?utm_campaign=homeAdvegas1?%20)), so về tỷ lệ cho tử vong là dương tính và âm tính phần còn lại có thể thấy được tỷ lệ ta nhận được cực kỳ mất cân bằng với 2.156% bệnh nhân tử vong và đây cũng là một vấn đề cực kỳ khó chịu trong việc xây dựng mô hình Machine Learning dựa trên những dữ liệu này. Tuy đây là một chủ đề tiêu cực trong việc phòng chống dịch khi tiên lượng tỷ lệ tử vong của bệnh nhân trong quá trình điều trị nhưng dưới góc nhìn khách ở một người nghiên trong lĩnh vực y học này ta cần phải thừa nhận rằng có một điều chắc chắn là không điều gì là chắc chắn, ngay cả khi bệnh nhân vẫn vui vẻ khỏe mạnh trong hôm nay nhưng khi đến ngày mai thì chưa chắc vậy. Vậy nên chủ đề này được sử dụng như là một dấu hiệu cảnh báo cho việc chăm sóc bệnh nhân của những lương y

Ta sẽ nói sơ về khái niệm của như sự ảnh hưởng và giải pháp, mất cân bằng lớp học là một vấn đề phổ biến trong thực hành Machine Learning có thể ảnh hưởng nghiêm trọng đến cả chức năng của thuật toán và lựa chọn mô hình. Điều này đề cập đến các tình huống khi các lớp không được phân phối đồng đều trong tập dữ liệu. Vấn đề mất cân bằng có thể gặp phải trong các nhiệm vụ phân loại 2 lớp hoặc đa lớp.

Hãy tưởng tượng nếu ta muốn xây dựng một mô hình Machine Learning để phát hiện một căn bệnh hiếm gặp hoặc một sự kiện lâm sàng hiếm gặp. Vì kết quả mục tiêu là rất hiếm, chỉ có 100 trường hợp dương tính hoặc ít hơn trong tập dữ liệu 1000 trường hợp. Sau khi phân tách dữ liệu, chúng tôi giữ một tập hợp con 10% để xác thực. Tập con thử nghiệm này có 10 bệnh nhân và 90 người khỏe mạnh. Tập huấn mô hình đề xuất hai mô hình: A và B. Mô hình A phân loại nhầm 6/10 bệnh nhân là âm tính (âm tính giả) và 10/90 đối tượng khỏe mạnh là dương tính (dương tính giả) trong khi mô hình B chỉ phân loại sai 2/10 bệnh nhân là khỏe mạnh và 30 / 90 người bình thường như mắc bệnh. Dựa trên độ chính xác tuyệt đối và tỷ lệ lỗi tuyệt đối, máy tính có thể coi rằng mô hình A tốt hơn mô hình B; vì nó chỉ mắc 16 sai lầm so với 32 sai lầm của mô hình B. Tuy nhiên, chúng ta nên chọn mô hình B để thay thế, vì lỗi âm tính giả nguy hiểm hơn dương tính giả. Vấn đề là chúng ta thậm chí có thể không có cơ hội để suy luận như vậy, vì mô hình B đã bị máy tính loại bỏ trong quá trình đào tạo. Một số thuật toán chỉ hoạt động tốt nhất trên dữ liệu cân bằng. Đối với vấn đề phân loại nhiều lớp, quá trình đào tạo cũng có thể bị gián đoạn do mất cân bằng lớp.

Một chiến thuật phổ biến để giảm ảnh hưởng xấu của sự mất cân bằng các lớp là kiểm soát quá trình Resampling. Việc kiểm soát có thể bao gồm Oversampling (bù cho lớp thiểu số bằng cách sao chép dữ liệu) hoặc [Undersampling](file:///C:\Users\Hendrich\Desktop\Professional%20Internship\Doc\Category\Undersampling.docx) (loại bỏ các thể hiện khỏi lớp đa số). Nhưng không cách nào trong số đó thực sự tốt. Bằng cách Resampling một số trường hợp thông tin có thể bị mất, trong khi Oversampling c có thể dẫn đến hiện tượng Overfitting . Chúng ta cũng có thể kết hợp Oversampling và Undersampling với sự cân bằng giữa tác động tích cực và tiêu cực của chúng.

H2o cung cấp một giải pháp đơn giản và linh hoạt để đối phó với sự mất cân bằng dữ liệu dựa trên điều khiển lấy mẫu lại, thông qua hai tham số trong quá trình đào tạo: Các Balances-class và Fold-Asigment. Hai tham số đó chỉ có hiệu lực trên Random Forest, Deep neuralnet và GBM, cũng là những Machine Learning liên quan đến việc tách ngẫu nhiên. Các tham số có thể được đặt trong quá trình huấn luyện mô hình nhưng không được đặt trên khung dữ liệu.

Khi thông số Balance-class được đặt là true, điều khiển lấy mẫu kết hợp được kích hoạt, điều này cho phép Oversampling hoặc Undersampling các nếp gấp huấn luyện dựa trên tỷ lệ các lớp trong khung huấn luyện ban đầu. Mỗi hàng ở các lớp đa số sẽ có trọng số cao hơn một hàng ở các lớp thiểu số. Người dùng cũng có thể đặt điều khiển rõ ràng cho tham số này bằng cách đưa các giá trị trọng số vào tham số class\_sampling\_factor (được khuyến nghị).

Folder\_assignment xác định cách phân chia dữ liệu đào tạo trong quá trình Cross-validations. Giá trị của nó có thể là “AUTO” (có nghĩa là phân chia ngẫu nhiên), Modulo và Stratifield. Khi được thiết lập là “Stratifielded” chương trình sẽ cố gắng “Balance” các lớp như nhau vào từng phần. Điều này có thể cải thiện đáng kể hiệu suất của mô hình trên dữ liệu không cân bằng.

Và như mô hình trên nên ta sẽ chỉ thay data rồi tiến hành xây dựng mô hình vì các bước EDA đã được xử lý ở mục trên. Đoạn code sau sẽ là một đoạn code gộp nó sẽ bao gồm các hàm call các library , import data, divide data, đồng thời cũng xem nhanh qua mô tả của dữ liệu.

|  |
| --- |
| library(tidyverse)  library(tibble)  library(explore)  library(dataMaid)  library(summarytools)  Sys.setenv(JAVA\_HOME="D:\\ZZZ Program Files\\Java")  library(rJava)  setwd("C:/Users/Hendrich/Desktop/Professional Internship/Death")  library(keras)  library(dplyr)  require(foreign)  require(farff)  library(ggplot2)  library(tidyr)  library(tibble)  df=read.arff("Death.arff")  makeDataReport(df,  render = FALSE,  file = "EDA.Rmd",  replace = TRUE)  names(df)=c("Diagnosis","FVC","FEV1","Zubrod",  "Pain","Haemoptysis","Dyspnoea","Cough","Weakness",  "T\_grade","DBtype2","MI","PAD","Smoking","Asthma","Age","Survival")  view(df)  df$Survival=df$Survival%>%recode\_factor(.,`F` = "Survived", `T` = "Dead")  df$Tiffneau=df$FEV1/df$FVC  Hmisc::describe(df) |

|  |
| --- |
| df  18 Variables 470 Observations  --------------------------------------------------------------------------------------------  Diagnosis  n missing distinct  470 0 7  lowest : DGN1 DGN2 DGN3 DGN4 DGN5, highest: DGN3 DGN4 DGN5 DGN6 DGN8    Value DGN1 DGN2 DGN3 DGN4 DGN5 DGN6 DGN8  Frequency 1 52 349 47 15 4 2  Proportion 0.002 0.111 0.743 0.100 0.032 0.009 0.004  --------------------------------------------------------------------------------------------  FVC  n missing distinct Info Mean Gmd .05 .10 .25 .50  470 0 134 1 3.282 0.9818 2.018 2.316 2.600 3.160  .75 .90 .95  3.808 4.560 4.900  lowest : 1.44 1.46 1.70 1.81 1.82, highest: 5.52 5.56 5.60 6.08 6.30  --------------------------------------------------------------------------------------------  FEV1  n missing distinct Info Mean Gmd .05 .10 .25 .50  470 0 136 1 4.569 4.805 1.440 1.640 1.960 2.400  .75 .90 .95  3.080 3.762 4.311  lowest : 0.96 1.00 1.04 1.12 1.20, highest: 76.00 76.80 78.30 79.30 86.30    Value 1 2 3 4 5 9 52 61 64 66 67 69 71  Frequency 27 224 151 47 6 1 1 1 1 1 1 1 1  Proportion 0.057 0.477 0.321 0.100 0.013 0.002 0.002 0.002 0.002 0.002 0.002 0.002 0.002    Value 73 76 77 78 79 86  Frequency 2 1 1 1 1 1  Proportion 0.004 0.002 0.002 0.002 0.002 0.002  For the frequency table, variable is rounded to the nearest 1  --------------------------------------------------------------------------------------------  Zubrod  n missing distinct  470 0 3    Value PRZ0 PRZ1 PRZ2  Frequency 130 313 27  Proportion 0.277 0.666 0.057  --------------------------------------------------------------------------------------------  Pain  n missing distinct  470 0 2    Value F T  Frequency 439 31  Proportion 0.934 0.066  --------------------------------------------------------------------------------------------  Haemoptysis  n missing distinct  470 0 2    Value F T  Frequency 402 68  Proportion 0.855 0.145  --------------------------------------------------------------------------------------------  Dyspnoea  n missing distinct  470 0 2    Value F T  Frequency 439 31  Proportion 0.934 0.066  --------------------------------------------------------------------------------------------  Cough  n missing distinct  470 0 2    Value F T  Frequency 147 323  Proportion 0.313 0.687  --------------------------------------------------------------------------------------------  Weakness  n missing distinct  470 0 2    Value F T  Frequency 392 78  Proportion 0.834 0.166  --------------------------------------------------------------------------------------------  T\_grade  n missing distinct  470 0 4    Value OC11 OC12 OC13 OC14  Frequency 177 257 19 17  Proportion 0.377 0.547 0.040 0.036  --------------------------------------------------------------------------------------------  DBtype2  n missing distinct  470 0 2    Value F T  Frequency 435 35  Proportion 0.926 0.074  --------------------------------------------------------------------------------------------  MI  n missing distinct  470 0 2    Value F T  Frequency 468 2  Proportion 0.996 0.004  --------------------------------------------------------------------------------------------  PAD  n missing distinct  470 0 2    Value F T  Frequency 462 8  Proportion 0.983 0.017  --------------------------------------------------------------------------------------------  Smoking  n missing distinct  470 0 2    Value F T  Frequency 84 386  Proportion 0.179 0.821  --------------------------------------------------------------------------------------------  Asthma  n missing distinct  470 0 2    Value F T  Frequency 468 2  Proportion 0.996 0.004  --------------------------------------------------------------------------------------------  Age  n missing distinct Info Mean Gmd .05 .10 .25 .50  470 0 45 0.998 62.53 9.737 49.45 52.00 57.00 62.00  .75 .90 .95  69.00 74.00 77.00  lowest : 21 37 38 39 40, highest: 78 79 80 81 87  --------------------------------------------------------------------------------------------  Survival  n missing distinct  470 0 2    Value Survived Dead  Frequency 400 70  Proportion 0.851 0.149  --------------------------------------------------------------------------------------------  Tiffneau  n missing distinct Info Mean Gmd .05 .10 .25 .50  470 0 406 1 1.47 1.484 0.5806 0.6375 0.7134 0.7738  .75 .90 .95  0.8307 0.8909 0.9859  lowest : 0.2659574 0.3934426 0.4262295 0.4653465 0.5000000  highest: 25.5893536 25.8503401 28.4400000 32.9047619 47.4175824    Value 0.5 1.0 3.0 16.0 17.5 21.0 23.5 24.0 25.0 25.5 26.0 28.5 33.0  Frequency 181 274 1 3 1 2 1 1 1 1 1 1 1  Proportion 0.385 0.583 0.002 0.006 0.002 0.004 0.002 0.002 0.002 0.002 0.002 0.002 0.002    Value 47.5  Frequency 1  Proportion 0.002  For the frequency table, variable is rounded to the nearest 0.5  -------------------------------------------------------------------------------------------- |

Các yếu tố dự báo bao gồm 16 biến:

Level là biến dự báo cấp độ và tình trạng của bệnh nhân DGN tăng từ DGN(2) nhỏ nhất tới DGN(8) là chuyến biến xấu nhất

FEV1,FVC lần lược là dung tích hơi thở trong 1 giây và thể tích khí có thể thở ra tối đa của bệnh nhân. Với thương của 2 đơn vị tren ta có chỉ số Tiffeneau-Pinelli là chỉ sổ đánh giá tình trạng suy hô hấp.

Điểm [Zubrod](file:///C:\Users\Hendrich\Desktop\Professional%20Internship\DocForReport%20(Medicine)\Zubord.docx) là điểm đánh giá trạng thái của bệnh nhân trong quá trình điều trị.

Pain là trạng thái tiêu cực được thể hiện khi bệnh nhân xuất hiện hiện tượng đau đớn trong quá trình điều trị.(True/False)

Haemoptysis là hiện tượng bệnh nhân xuất hiện xuất huyết trong quá trình điều (True/False)

Sultry là biến đã được sử dụng trên bài trước nó thể hiện cho việc bệnh nhân xuất hiện tình trạng suy hô hấp.(True/False)

Cought thể hiện cho hiện tượng ho của bệnh nhân (True/False)

Weakness khi bệnh cơ thể của bệnh nhân xuất hiện các dấu hiệu, triệu chứng suy nhược. (True/False)

T\_grade là biến thể hiện kích thước vùng consolidation trong phổi của bệnh nhân trong qua trình điều trị (từ đơn vị OCT11 đến OCT13)

Age thể hiện tuổi của bệnh nhân.(Integer)

Asthma là biến dược thể hiện bệnh hen xuyễn (True/ False)

Smoking là biến được thể hiện khi bệnh nhân đã từng dụng thuốc lá.(True/False).

Và các biến MI, Dbtype2, PAD là các biến đa chuẩn đoán nghĩa là bệnh nhân đã hoặc đang mắc các bệnh khác trong quá trính điều trị hoặc trước quá trình điều trị. Với ý nghĩa lần lượt của các biến là MI là bệnh nhồi máu cơ tim, DBType2 là bệnh tiểu đường loại 2 và PAD là bệnh giảm động mạch ngoại biên. (True/False).

Kết quả mục tiêu rất mất cân bằng với tỷ lệ 400 người sống sót / 70 người chết (khoảng 1: 6). Tại sao lại có sự chêch lệch như vậy và lý do tại sao mô hình lại đưa ra dự đoán một các phi logic như thế? Để tìm câu trả lời cho câu hỏi này chúng hãy khám phá một chút về mô hình và vấn đề chính của ta chính là tìm ra những luận điểm và tính chất của dữ liệu khiến mô hình đạt được một chuẩn đoán như vậy. Như tôi đã nói phần giới thiệu “muốn áp dụng bất cứ thứ gì bạn cần hiểu tính chất và quy trình hoạt động cụ thể của nó”. Thứ tôi đang muốn nói đó chính là tính tường minh của một mô hình, một mô hình có thể là một thể là thứ gì đó “bí mật” nhưng nó không thể là một thứ “bí ẩn”.

Dựa trên những biến này tôi tiến hành xây dựng mô hình bằng cách khởi động package h2o và tạo mô hình với các siêu tham số đã nói ở trên cùng một vài tham số quan trọng khác

|  |
| --- |
| library(h2o)  h2o.init(nthreads = -1,max\_mem\_size ="1g")  wtrain=as.h2o(trainset)  wtest=as.h2o(testset)  response="Survival"  features=setdiff(colnames(wtrain),response)  # Balance Class and Stratifield  rfmod1=h2o.randomForest(x = features,  y = response,  training\_frame = wtrain,nfolds=10,  fold\_assignment = "Stratified",  balance\_classes = TRUE,class\_sampling\_factors=c(1.17,6.66),  ntrees = 100, max\_depth = 50,  stopping\_metric = "logloss",  stopping\_tolerance = 0.01,  stopping\_rounds = 3,  keep\_cross\_validation\_fold\_assignment = TRUE,  keep\_cross\_validation\_predictions=TRUE,  score\_each\_iteration = TRUE,  seed=12345)  h2o.performance(rfmod1,wtrain) |

Kiểm tra trên tập training ta thấy mô hình hoạt động rất tốt

|  |
| --- |
| H2OBinomialMetrics: drf  MSE: 0.007124389  RMSE: 0.08440609  LogLoss: 0.05032097  Mean Per-Class Error: 0  AUC: 1  AUCPR: 1  Gini: 1  R^2: 0.943704  Confusion Matrix (vertical: actual; across: predicted) for F1-optimal threshold:  Dead Survived Error Rate  Dead 55 0 0.000000 =0/55  Survived 0 315 0.000000 =0/315  Totals 55 315 0.000000 =0/370  Maximum Metrics: Maximum metrics at their respective thresholds  metric threshold value idx  1 max f1 0.617575 1.000000 66  2 max f2 0.617575 1.000000 66  3 max f0point5 0.617575 1.000000 66  4 max accuracy 0.617575 1.000000 66  5 max precision 1.000000 1.000000 0  6 max recall 0.617575 1.000000 66  7 max specificity 1.000000 1.000000 0  8 max absolute\_mcc 0.617575 1.000000 66  9 max min\_per\_class\_accuracy 0.617575 1.000000 66  10 max mean\_per\_class\_accuracy 0.617575 1.000000 66  11 max tns 1.000000 55.000000 0  12 max fns 1.000000 192.000000 0  13 max fps 0.029153 55.000000 115  14 max tps 0.617575 315.000000 66  15 max tnr 1.000000 1.000000 0  16 max fnr 1.000000 0.609524 0  17 max fpr 0.029153 1.000000 115  18 max tpr 0.617575 1.000000 66  Gains/Lift Table: Extract with `h2o.gainsLift(<model>, <data>)` or `h2o.gainsLift(<model>, valid=<T/F>, xval=<T/F>)` |

Tuy nhiên, kiểm tra mô hình trên tập test lại cho kết quả tương đối bi quan: hình tiên lượng cho các bệnh nhân sống sót rất tốt nhưng nó lại sai hết trong việc tiên lượng cho bệnh nhân tử vong 60%.

|  |
| --- |
| h2o.performance(rfmod1,wtest) |

|  |
| --- |
| H2OBinomialMetrics: drf  MSE: 0.202316  RMSE: 0.4497955  LogLoss: 1.21167  Mean Per-Class Error: 0.5  AUC: 0.5768627  AUCPR: 0.8830669  Gini: 0.1537255  R^2: -0.5867922  Confusion Matrix (vertical: actual; across: predicted) for F1-optimal threshold:  Dead Survived Error Rate  Dead 2 6 0.6000000 =6/8  Survived 0 92 0.000000 =0/92  Totals 0 100 0.150000 =6/100  Maximum Metrics: Maximum metrics at their respective thresholds  metric threshold value idx  1 max f1 0.183272 0.918919 51  2 max f2 0.183272 0.965909 51  3 max f0point5 0.183272 0.876289 51  4 max accuracy 0.183272 0.850000 51  5 max precision 0.925916 0.926829 12  6 max recall 0.183272 1.000000 51  7 max specificity 1.000000 0.866667 0  8 max absolute\_mcc 0.925916 0.179365 12  9 max min\_per\_class\_accuracy 0.659247 0.533333 19  10 max mean\_per\_class\_accuracy 0.925916 0.623529 12  11 max tns 1.000000 13.000000 0  12 max fns 1.000000 67.000000 0  13 max fps 0.345395 15.000000 43  14 max tps 0.183272 85.000000 51  15 max tnr 1.000000 0.866667 0  16 max fnr 1.000000 0.788235 0  17 max fpr 0.345395 1.000000 43  18 max tpr 0.183272 1.000000 51  Gains/Lift Table: Extract with `h2o.gainsLift(<model>, <data>)` or `h2o.gainsLift(<model>, valid=<T/F>, xval=<T/F>)` |

Vậy tại sao lại như vậy, khởi đầu cho tính tường minh của mô hình tôi tiến hành phương pháp LIME với 2 lý đo chính. Thứ nhất như đã thống kê ở trên mô hình này thuộc dạng dữ liệu it nên việc diễn giải sẽ thuộc về việc tối ưu hóa ở mặt thời gian, thứ 2 là mô hình trên đã được diễn giải bằng phương pháp RandomExplainer rồi nên một phương pháp mới sẽ làm mọi việc hứng thú hơn

Đầu tiên, chúng tôi gọi tên các thư viện , sau đó viết 2 hàm model\_type.H2Omodel và predict\_model.H2Omodel, để thích nghi với đầu ra định dạng của mô hình h2O với bên trong chuẩn cấu trúc trong LIME:

Sau đó, ta tạo object explainer chứa phân phối của features từ training set và mô hình

|  |
| --- |
| library(lime)  # Adapting lime functions to h2O framework  model\_type.H2OModel<- function(x, ...) "classification"  predict\_model.H2OModel <- function(x, newdata, type, ...) {  pred <- h2o.predict(x, as.h2o(newdata))  return(as.data.frame(pred[,-1]))  }  set.seed(12345)  explainer <- lime(trainset[,-17], rfmod2, bin\_continuous = FALSE, n\_bins = 10, n\_permutations = 10000) |

Tiếp theo, ta tách riêng ra 2 phần testset: những case dương tính và những case âm tính.

|  |
| --- |
| library(magrittr)  library(dplyr)  deaddf=subset(testset,Survival!="Survived")%>%.[,-17]  survdf=subset(testset,Survival=="Survived")%>%.[,-17] |

Việc lựa chọn M features được thực hiện bằng nhiều cách, có thể tùy chọn, thí dụ :

None : không lựa chọn ưu tiên, mà tính weights score cho toàn bộ feature trong model, cách này không được khuyến khích nếu M>10

Foward selection (hay ridge regression) : Mô hình tuyến tính được dựng theo ridge regression, có khuynh hướng giảm tham số hồi quy về gần 0 và loại bỏ những feature không quan trọng

Highest weights : Chọn m feature với giá trị weight tuyệt đối cao nhất, dựa vào ridge regression

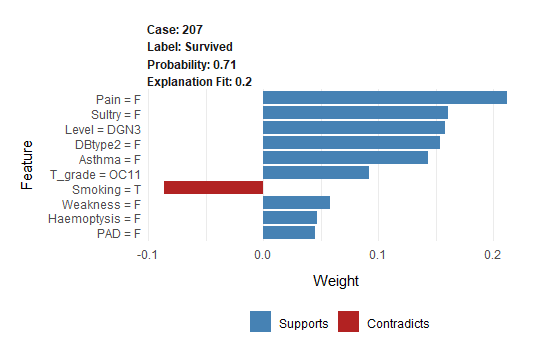
Lasso : Dựng mô hình tuyến tính bằng phương pháp Lasso, nó giúp loại bỏ triệt để những feature không cần thiết bằng cách đẩy weight score về 0.

Tree: dựng mô hình decision tree được phân cấp dựa vào log2(M) từ giá trị M cao nhất.

Auto: tự động, lime dùng phương pháp forward nếu m<=6 hoặc highest weights nếu m>6.

Đầu tiên, chúng ta chọn ngẫu nhiên 1 bệnh nhân (số 49) trong nhóm bệnh viêm phổi, và giải thích tại sao mô hình lại chuẩn đoán bệnh nhân này là bị viêm phổi.

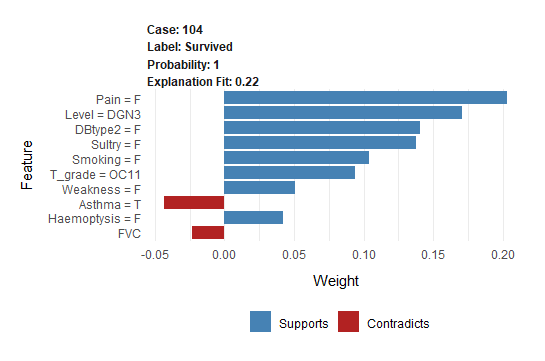
|  |
| --- |
| require(caret)  caseS1<-lime::explain(survdf[15,], explainer, labels="Survived",  n\_features =10,feature\_select="auto")  plot\_features(caseS1) |



Kết quả cho thấy rằng: trong 10 features có ảnh hưởng cao nhất đến tiên lượng của bệnh nhân có đến 9 yếu tố ủng hộ cho kết quả sống sót, bệnh nhân không có triệu chứng đau đớn, không khó thở, không mắc bệnh tiểu đường, không hen xuyễn, không suy nhược ,không thổ huyết và không mắc triệu chứng giảm động mạch ngoại biên. Tuy nhiên, có một feature mâu thuẫn đó là bệnh nhân có hiện tượng sử dụng thuốc lá.

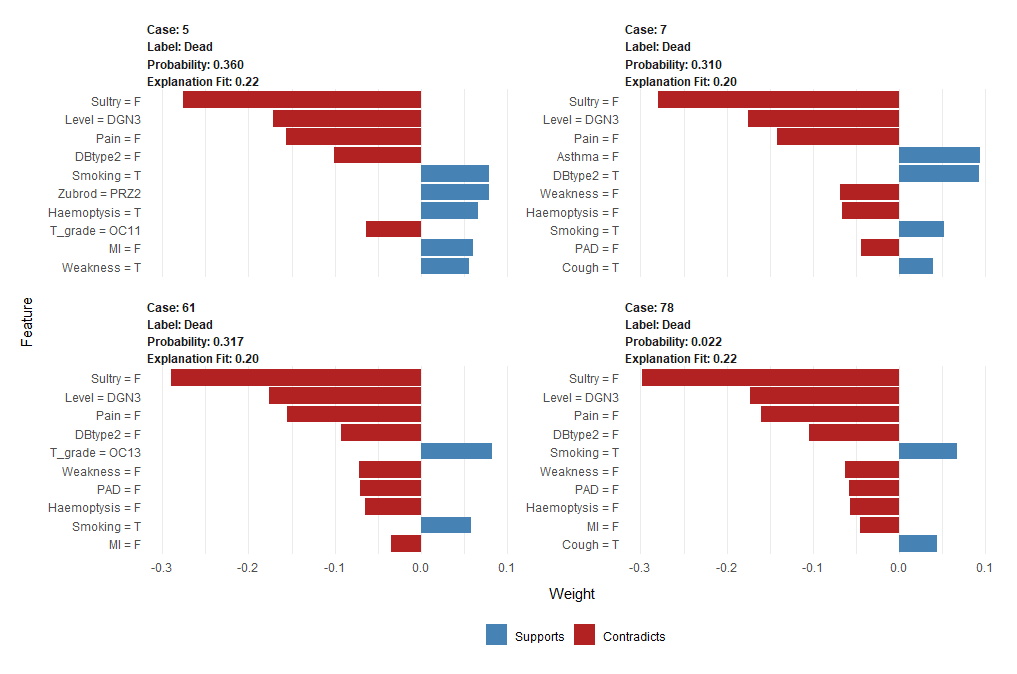
Xem thêm trường hợp khác

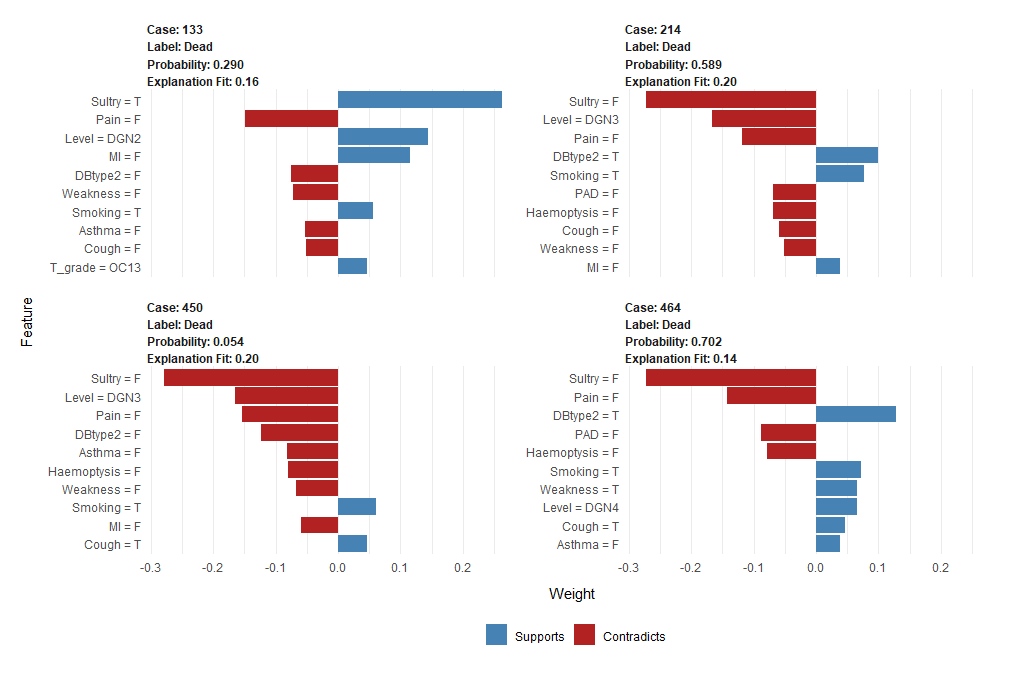
|  |
| --- |
| case2 <- lime::explain(survdf[8,], explainer = explainer, labels = "Survived",  n\_features = 10, feature\_select = "auto")  plot\_features(case2) |



Ở bệnh nhân này ta thấy được bệnh nhân có 8/10 feature ủng hộ cho việc sống sót và nó đem lại xác xuất tuyệt đôi trừ 2 chi tiêt mâu thuẫn là bệnh nhân có dấu hiệu bị suy nhược cơ thể và dung tích thở thấp hơn ngưỡng an toàn. Như vậy chúng ta đã tạm thời giải thích được tại sao mô hình tiên lượng sống sót lại chính xác, bây giờ ta sẽ thử giải thích tại sao nó lại tiên lượng tử vong sai? Vì tập test chỉ có 8 nên tôi sẽ chia nó thành 2 phần với mỗi phần là 8 biểu đồ tương ứng

|  |
| --- |
| dead1<-lime::explain(deaddf[c(1:4),], explainer, labels="Dead",  n\_features =10,feature\_select="auto")  plot\_features(dead1)  dead2<- lime::explain(deaddf[c(5:8),], explainer, labels = "Dead",  n\_features = 10 , feature\_select = "auto")  plot\_features((dead2)) |





Dựa vào thống kê trên ta có thể thấy được rằng chỉ có đúng 2 trường hợp tiên lượng dương tính thật, tương đối chính xác dựa trên hệ nhị phân (bệnh nhân số 214 xác xuất tử vong ~ 59% và bệnh nhân 464 xác xuất tử vong đạt tỷ lệ 70%); còn lại kết quả tiên lượng là âm tính giả (nghĩa là bệnh nhân tử vong mà mô hình không tiên lượng được). Xem xét lại những biến ủng hộ và mâu thuẫn trong cả 2 trường hợp, ta thấy chúng đều có ích, từ đó ta có thể suy luận cơ chế mà mô hình hoạt động và nhận ra một vài điểm bất thường của kết quả trên:

Đầu tiên đặc tính hút thuốc lá là một biến quan trọng, nó ủng hộ cho việc tiên lượng bệnh nhân tử vong một cách có logic bằng một cách thông thường nhất ta có thể thấy được rằng tất cả trường hợp tử vong đều có đặc tính sử dụng thuốc lá và đặc biệt là trong 2 trường hợp được tiên lượng chính xác bên trên cũng đều xử dụng thuốc lá. Giải thích dưới góc nhìn của bác y khoa ta có thể thấy rằng việc hút thuốc được cấu tạo và hình thành dựa trên 5.000 chất hóa học độc hại khác nhau việc hút thuốc sẽ khiến các phế nang làm việc liên tục, thêm vào đó việc khói thuốc sẽ làm viêm các phế nang là tắc nghẽn khí huyết trong quá trình cung cấp khí từ đó dẫn tới việc bệnh nhân mắc thêm những chứng bệnh nghiêm trọng về hô hấp cộng thêm việc nhiễm một bệnh suy hô hấp nghiêm trọng như COVID thì tỷ lệ tử vong của các bệnh nhân này sẽ là cực kỳ cao. Tuy nhiên 1 mình features smoking thì không đủ để kéo outcome theo hướng positive, vì nó bị các triệu chứng khác gây nhiễu. Có vẻ như sự vắng mặt của các triệu chứng không đủ để xác định nguy cơ tử vong, trong 8 bệnh nhân tử vong hầu hết đều không có các dấu hiệu nguy cấp như là khó thở, suy nhược cơ thể hay ho ra máu nhưng họ vẫn chết. Dung tích hơi thở và dung tích phổi của nạn nhân trong trường hợp này cũng không phải là một features có giá trị.

Nói một cách khác mô hình Random Forest đã không thành công trong việc phân tích và tiên lượng sống sót của bệnh nhân dựa trên những triệu chứng này. Mặc dù đã cố gắng phân tích khuyết điểm mô hình nhưng tôi vẫn không thể cải thiện chất lượng của và nguyên nhân xuất phát mà theo tôi nghĩ đến là: Việc mất cân bằng lớp học và việc dữ liệu quả ít dẫn đến tình trạng mô hình bị Overfitting. Về mất cân bằng lớp như đã nói ở trên đây là một vấn đề phổ biến và khó chịu trong thực hành học máy. Balanace class là một tính năng hữu ích trong h2o có thể được sự dụng kết hợp với Fold-assignment để giải quyết vấn đề trên nhưng còn vấn đề Overfitting mặc dùng cố gắng sử dụng phương pháp Cross-validation nhưng vẫn không thể cải thiện được xác xuất nên lựa chọn tối ưu nhất mà tôi có thể nghĩ đến là tăng kích thước tập dữ liệu.

# CHƯƠNG III: KẾT LUẬN VÀ ĐÁNH GIÁ

1. **Đánh giá tổng quan.**

Qua bài báo cáo trên hi vọng mọi người có một cái nhìn toàn diện về sự quan trọng của tính tương mình trong một thuật toán và sự tiện ích mà Machine Learning mang lại trong các ngành nghề đặc biệt là những ngành nghề cần tốc độ xử lý nhanh và tính chính xác cao nhưng phải tường minh trong từng giai đoạn xử. Từ 2 mô hình trên ta có thể rút ra một vài điều quan trọng:

* Dữ liệu là thứ quan trọng và tất yếu. đây là thứ tuyệt mật, nhiều công ty sẵn sàng công khai thuật toán mà họ sử dụng để xây mô hình nhưng dữ liệu thì không nên mong muốn dữ liệu trên được xây dựng trên mô hình một không được tuồn ra ngoài phạm vi của bài báo cáo này.
* Trước khi xây dựng mô hình bước đầu tiên cần phải xem xét dữ liểu để biết ta đang có cố gắng xây dựng cái gì? Trong khi ta có cái gì? Và mục tiêu của mô hình xây dựng ra gì? Đôi khi cũng là này xuất hiện ở đâu trong mô hình?
* Trong các trường hợp dự đoán lớp nhị phân (tức là lớp chỉ có 2 trường hợp) với 1 là dương tính và 0 là âm tính. Mặc dù hơi mang tính y học nhưng ta phải cực kỳ chú ý đến các trường hợp âm tính giải, đây là những trường hợp vô cùng nguy hiểm cần phải được xử lý theo nguyên tắc “*Thà giết lầm còn hơn bỏ sót*”.

1. **Ưu điểm và nhược điểm.**

### **Ưu điểm.**

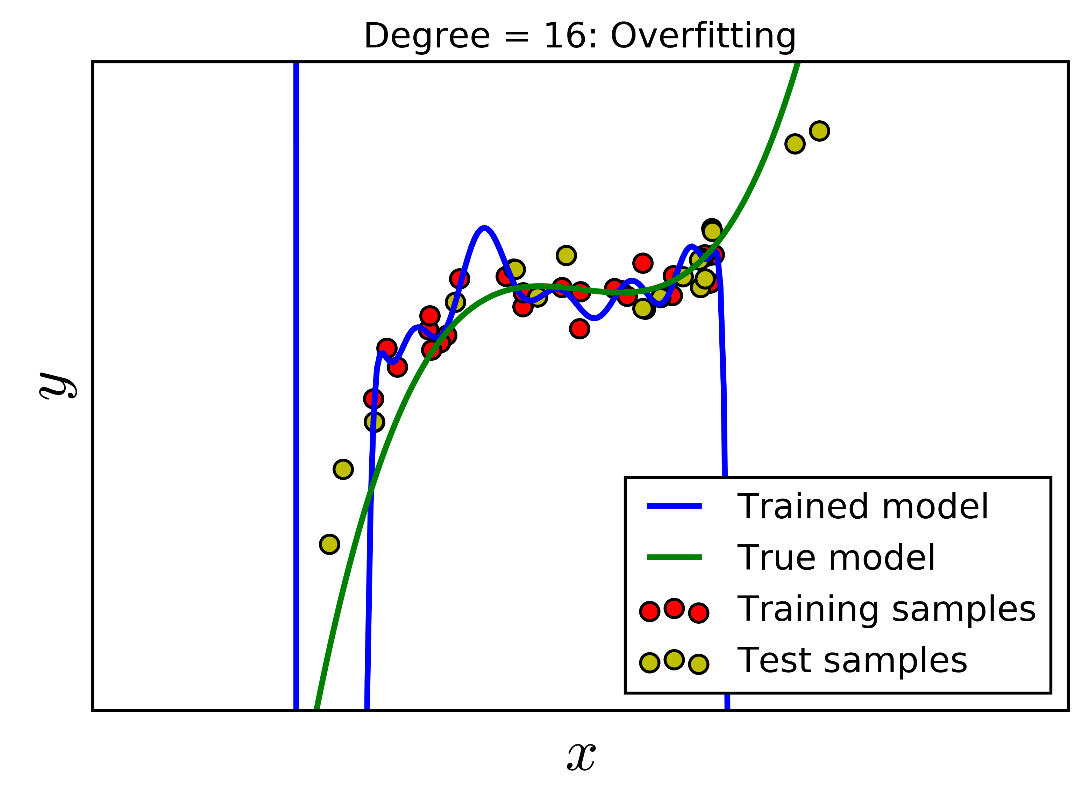
* Giải nghĩa được cơ chế hoạt động của mô hình thuật toán
* Cũng cố tính tường minh của machine learning trong vấn đề y học

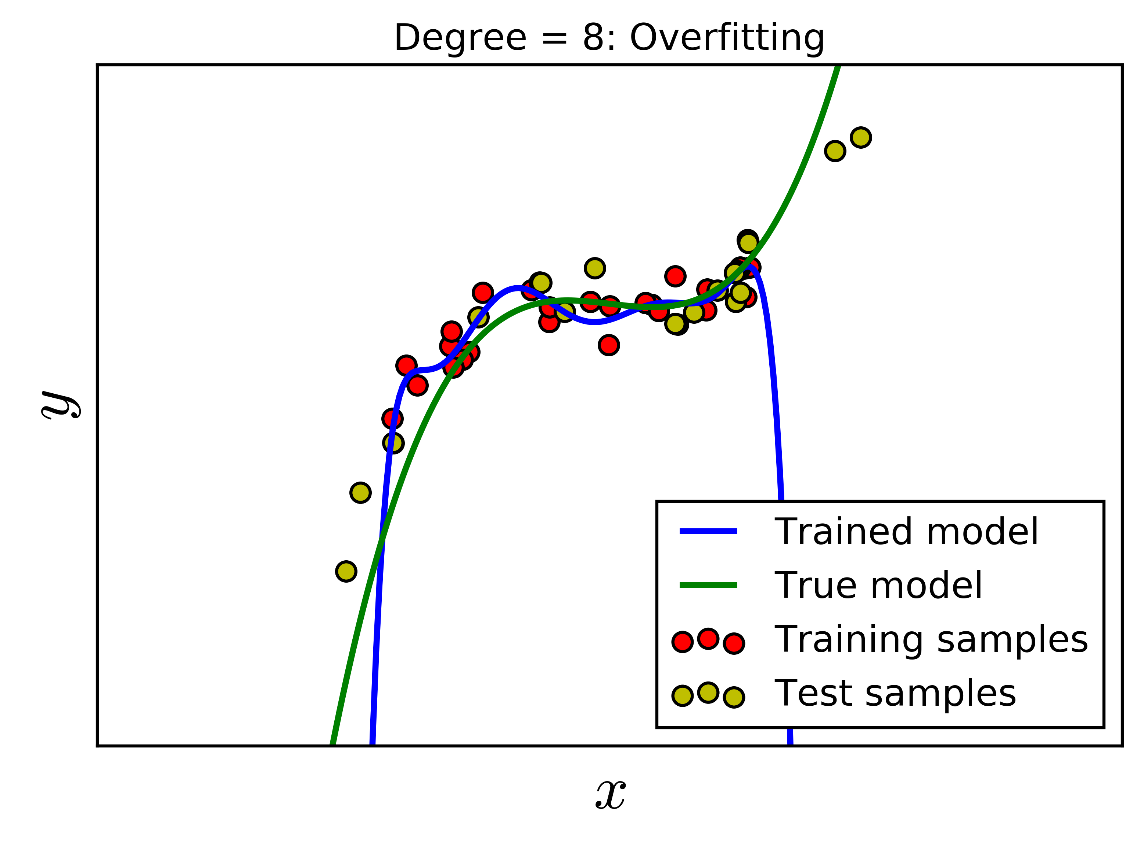
### **Nhược điểm.**

* Mô hình đưa ra tỷ lệ chính xác tương thấp khó chấp nhận trong y học
* Dữ liệu còn thiếu sót chưa nhiều từ đó ảnh hưỡng đến vấn đề chuẩn đoán của mô hình
* Xảy ra một vài trường hợp khách quan từ phía bệnh nhân khiến dữ liệu bị nhiễu
* Là ý tưởng cho việc áp dụng Machine Learning vào y học với độ tường minh cao nhưng khả năng áp dụng được vào trong y học thực tế là chưa cao
* Cần tìm giải pháp tối ưu hơn cho vấn đề dữ liệu

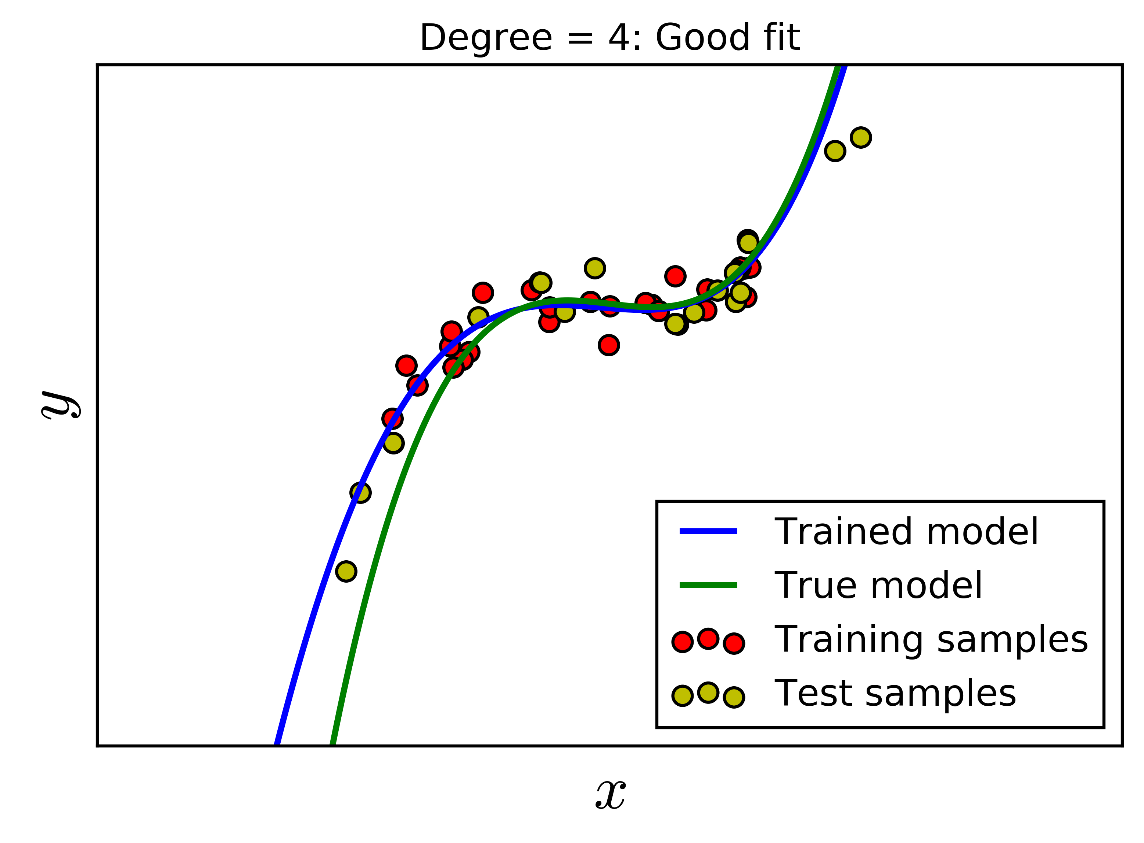
## **Hướng phát triển.**

* Như được nói ở phần khuyết điểm, cần tìm giải pháp tối ưu hơn cho vấn đề dữ liệu, đồng thời đây cũng được coi làm một câu trả lời thích đáng cho câu hỏi được nêu trong bài báo cáo. Nhắc lại câu hỏi một chút “*Tại sao* *Có một vài trường hợp giữa 2 bệnh nhân, tất cả triệu chứng giữa 2 họ đều giống nhau nhưng sau khi có kết quả xét nghiệm một trong số họ lại dương tính người còn lại là âm tính ?*”. đây là một câu hỏi thú vị được phát hiện trong lúc tôi tiến hành EDA dữ liệu và đi kèm sau câu trả lời này sẽ là phương thức áp dụng dữ liệu mới, tối ưu hơn chính xác hơn. trực quan hơn nhưng đồng thời cũng khó khăn hơn. Và câu trả lời cho vấn đề này được gọi là Đa chuẩn đoán (y học) và bệnh lồng bệnh (theo cách gọi thông thường). Đa chuẩn đoán là hiện tượng người bệnh được chuẩn đoán mắc hoặc nhiễm 2 bệnh trong cùng một lúc nghĩa là trong khi học mắc bệnh này đồng thời họ lại mắc một căn bệnh khác, như vậy áp dụng nguyên lý kế thừa người bệnh sẽ thừa hưởng cả hai triệu chứng của 2 loại bệnh nhưng lại chỉ đứng tên bởi một loại bệnh. Có thể nói đây là một điều cực kỳ tệ hại trong y học bởi nó gây ra một vùng nhiễu dữ liệu cực mạnh và cực khó để giải quyết đó là lý do khi nhìn vào biểu đố đường cong học tập, ta thấy dữ liệu luôn ở mức tạm ổn vì tỷ lệ lỗi luôn giao động, một mô hình khi được xây dựng trên dữ liệu này thì quá trình đạo tạo sẽ cực kỳ mất thời gian (4 tiếng cho demo bên trên không bao gồm việc nhập lệnh) bởi nó quá phức tạp để huấn luyện một hình khi dữ liệu bị nhiễu. Như vậy từ đó ta có thể thấy một điều là dữ liệu trên đã bị nhiễu và tỷ lệ mà nó được ra nếu được tối ưu hóa nhất nó sẽ mang thiên hướng “*Brute-force*” nghĩa là nó sẽ loại trừ tất cả các trường hợp mà nó cho âm tính và tập trung quét mạnh vào trường hợp dương tính từ đó chỉ cần bệnh nhân có một trong những dấu hiệu như khó thở hoặc ho thì mô hình vẫn sẽ chuẩn đoán là họ đã nhiễm COVID 19. Nhưng một mô hình tốt là mô hình chính xác cho cả 2 trường hợp nghĩa là nó sẽ tiến hành quét hết những trường hợp mang những đặc trung phổ biến vào đúng lớp nó thuộc về, để làm nhưng vậy ta cần có một bộ dữ liệu với kiểu dữ liệu mới đó chính là hình ảnh, với ảnh Xquang ta có thể xác định được nhiều thứ trong khu vực giới hạn mà ta cần từ đó đưa ra một sự chuẩn đoán cực kỳ chính ở các 2 trường hợp.
* Với tập dữ liệu thứ 2 để tăng độ chính xác của mô hình ta chỉ còn cách tăng số lượng tập dữ liệu training để mô hình không xuất hiện tình trạng Overfitting.





Và mô hình tối ưu nhất được thể hiện



* Trong quá trình kết thúc bài cáo tôi có đồng thời tham gia vào trường hè FIT được tổ chức bởi trường Đại hoc Khoa Học Tự Nhiên Đại học quốc gia TP Hồ Chí Minh, tại đây tôi đươc học rất nhiều kiến thức khá bổ ích trong việc áp dụng machine learning cho các vấn đề phức tạp điển hình CNN và RNN trong bài toán nhận diện hình ảnh. Đây là

1. **Lời Kết**

Như vậy bài cáo này là đại diện cho quá trình tìm hiểu và nghiên cứu của tôi về những kiến thức chuyên ngành và những kiến thúc ngoài chuyên ngành trong thời gian thực tập chuyên môn. Đồng thời đây cũng một bước xác định và định hướng nghiên cứu trong tương lai khi kết thúc bài báo cáo này ta lại rút ra những khuyết điểm thứ mà có thể khắc phục và cải thiện trong bài báo cáo tới.

Xin Cảm Ơn Quý Thầy cô đã xem xét.

**Tài liệu tham khảo**

* [**https://www.nhs.uk/conditions/coronavirus-covid-19/symptoms/**](https://www.nhs.uk/conditions/coronavirus-covid-19/symptoms/)
* [**https://towardsdatascience.com/understanding-random-forest-58381e0602d2**](https://towardsdatascience.com/understanding-random-forest-58381e0602d2)
* [**https://www.edureka.co/blog/classification-in-machine-learning/#:~:text=MNIST%20Digit%20Classification-,What%20is%20Classification%20In%20Machine%20Learning,as%20target%2C%20label%20or%20categories**](https://www.edureka.co/blog/classification-in-machine-learning/#:~:text=MNIST%20Digit%20Classification-,What%20is%20Classification%20In%20Machine%20Learning,as%20target%2C%20label%20or%20categories)**.**
* [**https://machinelearningmastery.com/k-fold-cross-validation/**](https://machinelearningmastery.com/k-fold-cross-validation/)
* [**https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/algo-params/balance\_classes.html**](https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/algo-params/balance_classes.html)
* [**http://hoitho-cuocsong.org.vn/wp-content/uploads/2016/10/7.-XQUANG-NGUC-CO-BAN-2.pdf**](http://hoitho-cuocsong.org.vn/wp-content/uploads/2016/10/7.-XQUANG-NGUC-CO-BAN-2.pdf)
* [**https://homes.cs.washington.edu/~marcotcr/blog/lime/**](https://homes.cs.washington.edu/~marcotcr/blog/lime/)
* [**https://medium.com/xebia-france/explain-your-ml-model-predictions-with-local-interpretable-model-agnostic-explanations-lime-82343c5689db**](https://medium.com/xebia-france/explain-your-ml-model-predictions-with-local-interpretable-model-agnostic-explanations-lime-82343c5689db)
* [**https://cran.r-project.org/web/packages/randomForestExplainer/randomForestExplainer.pdf**](https://cran.r-project.org/web/packages/randomForestExplainer/randomForestExplainer.pdf)
* [**https://lgatto.github.io/IntroMachineLearningWithR/an-introduction-to-machine-learning-with-r.html**](https://lgatto.github.io/IntroMachineLearningWithR/an-introduction-to-machine-learning-with-r.html)
* [**https://translate.google.com/**](https://translate.google.com/)
* [**https://stackoverflow.com/**](https://stackoverflow.com/)
* [**https://machinelearningcoban.com/**](https://machinelearningcoban.com/)