TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN MÔN HỆ THỐNG THƯƠNG MẠI THÔNG MINH

DỰ BÁO THỜI TIẾT KHÍ HẬU VÀ LƯU LƯỢNG MƯA Ở AUSTRALIA

Người hướng dẫn: **TS DƯƠNG HỮU PHÚC**

Người thực hiện: **NGUYỄN HOÀNG TRÚC – 51800319**

PHÙ Ý KỲ - 51800989

NGUYỄN VĂN ĐIỂM - 51900310

Lớp: **18050402 - 19050302**

Khóa: **22-23**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN MÔN HỆ THỐNG THƯƠNG MẠI THÔNG MINH

DỰ BÁO THỜI TIẾT KHÍ HẬU VÀ LƯU LƯỢNG MƯA Ở AUSTRALIA

Người hướng dẫn: **TS DƯƠNG HỮU PHÚC**

Người thực hiện: **NGUYỄN HOÀNG TRÚC – 51800319**

PHÙ Ý KỲ - 51800989

NGUYỄN VĂN ĐIỂM - 51900310

Lớp: **18050402 - 19050302**

Khóa: **22-23**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021

LỜI CẢM ƠN

Trong quá trình học tập và nghiên cứu đề tài "Tìm hiểu phân tích Australia ngày tiếp theo liệu có mưa" chúng em nhận được sự giúp đỡ và hướng dẫn của Thầy Dương Hữu Phúc, để hoàn thành bài đồ án này. Chúng em xin cảm ơn và bảy tỏ lòng biết ơn với Ban giám hiệu nhà trường Đại học Tôn Đức Thắng, khoa Công nghệ thông tin, các thầy cô giáo đã tham gia quản lý, giảng dạy và giúp đỡ em trong suốt thời gian học tập môn Hệ thống thương mại thông minh. Chúng em xin bày tỏ lòng biết ơn sâu sắc đến Thầy Dương Hữu Phúc – Người đã trực tiếp giảng dạy môn môn Hệ thống thương mại thông minh. Chúng em xin cảm ơn bạn bè, thầy cô đã hỗ trợ và giúp đỡ chúng em trong quá trình học tập.

ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của thầy Dương Hữu Phúc. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào chúng tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày tháng năm

Tác giả

(ký tên và ghi rõ họ tên)

Nguyễn Hoàng Trúc

Phù Ý Kỳ

Nguyễn Văn Điểm

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

.....
.....
.....
.....
.....

TP. Hồ Chí Minh, ngày tháng năm
(ký tên và ghi rõ họ tên)

Phần đánh giá của GV chấm bài

.....
.....
.....
.....
.....

TP. Hồ Chí Minh, ngày tháng năm
(ký tên và ghi rõ họ tên)

TÓM TẮT

Trong bày báo cáo này gồm có 4 chương:

Chương 1 nói về tổng quan đề tài, giới thiệu đề tài và phương pháp nghiên cứu với ứng dụng thực tế của đề tài. Dựa vào data weatherAUS.csv được thu thập trong thực tế từ kaggle.com trong Rain in Australia để dự đoán xác suất ngày tiếp theo ở Australia có mưa không. Tiến hành xử lý missing value, tiến hành phân tích data set bằng cách train và test, áp dụng lý thuyết và thực thi đề tài bằng thuật toán hồi quy Logistic (Logistic Regression), thuật toán Machine Learning rừng ngẫu nhiên (Random Forest Machine Learning), thuật toán Naive Bayes và thuật toán XGBoost, trực quan hóa dữ liệu, đánh giá đo lường và cho ra kết quả.

Chương 2 tổng quan giải thuật cơ sở lý thuyết của các thuật toán đề áp dụng vào dataset dự báo thời tiết bao gồm các thuật toán hồi quy Logistic (Logistic Regression), thuật toán Machine Learning rừng ngẫu nhiên (Random Forest Machine Learning), thuật toán Naive Bayes và thuật toán XGBoost. Mỗi thuật toán sẽ được nêu lý thuyết liên quan và ưu nhược điểm của từng thuật toán, khi áp dụng vào dataset dự báo thời tiết thì sẽ áp dụng những gì.

Chương 3 thực nghiệm bao gồm đặt tả dataset nêu ra cụ thể từng thuộc tính có tác dụng gì, các yếu tố ảnh hưởng đến dự đoán mưa vào ngày mai, thực hiện hóa bằng dồ thị tableau và code bằng ngôn ngữ python demo dataset dự đoán xác suất. Khi cho ra được kết quả dự đoán sẽ so sánh được thuật toán nào tối ưu hơn khi áp dụng dataset dự báo thời tiết ở nước Úc

Chương 4 kết luận những điều đã đạt được trong báo cáo, tóm tắt các ý chính trong báo cáo, định hướng nghiên cứu trong tương lai.

MỤC LỤC

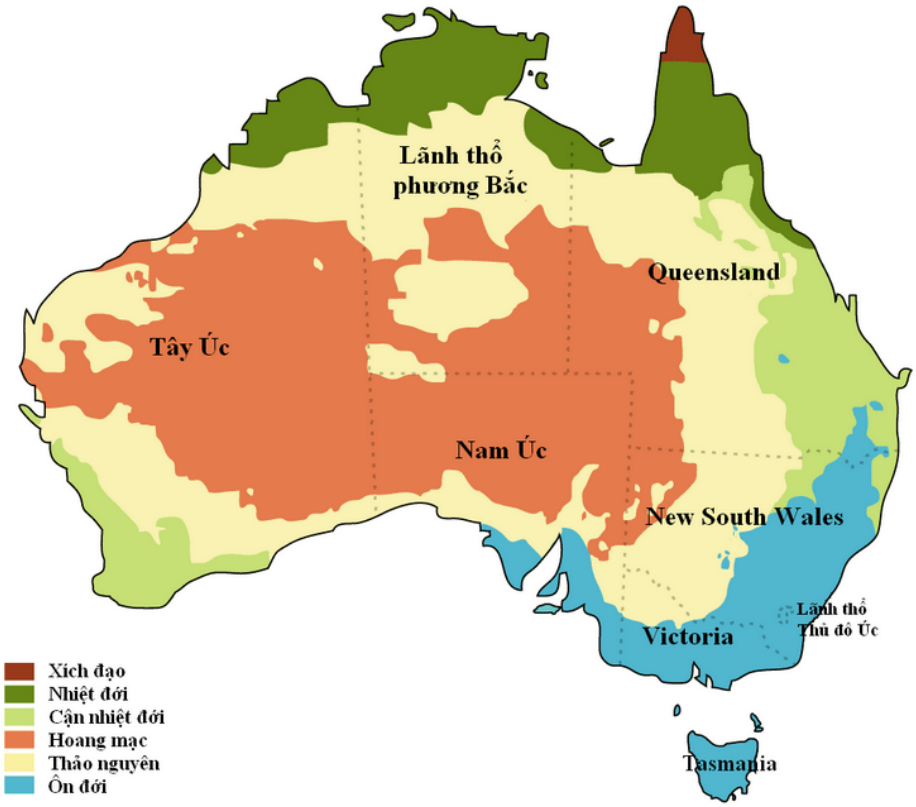
DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

Danh sách hình vẽ

Danh sách bảng

CHƯƠNG 1 - TỔNG QUAN ĐỀ TÀI

1.1 Giới thiệu đề tài



Hình 1: Bản đồ
(Nguồn:https://hchuman.com/thoi-tiet-nuoc-uc-khi-hau-4-mua-khac-la-o-cac-bang-cua-uc.html)

Nước Úc là nơi đa dạng về khí hậu. Để hiểu rõ nguyên nhân gây nên sự khác biệt của thời tiết nước Úc, chúng ta hãy quay về địa lý. Châu Úc nằm ở bán cầu Nam, đối lập thời tiết hoàn toàn với những nước ở bàn cầu Bắc. Khi ở bán cầu Bắc là mùa đông thì ở Úc là mùa hè ấm áp và ngược lại khi Úc chìm trong băng giá tháng 6 thì ở bán cầu Bắc đang là mùa hè rực rỡ.

Để thực hiên chính xác thì "Dự báo thời tiết" đã ra đời. Dự báo thời tiết là một trong những những nhu cầu thiết yếu mà con người vô cùng quan tâm và chú trọng bởi nó ảnh hưởng tới cuộc sống và kinh tế phát triển của quốc gia.

1.2 Phát biểu bài toán

Ngày nay vấn đề dự đoán thời tiết đã có vô số lời giải, có rất nhiều thuật toán được áp dụng trong lĩnh vực phân tích cho dự báo thời tiết hiệu quả bao gồm thuật toán hồi quy Logistic (Logistic Regression), thuật toán Machine Learning rừng ngẫu nhiên (Random Forest Machine Learning), thuật toán Naive Bayes và thuật toán XGBoost,...

Một trong những thuật toán được áp dụng trong lĩnh vực phân tích cho dự báo thời tiết hiệu quả nhất đó là Logistic Regression. Hồi quy logistic (Logistic Regression) là một mô hình thống kê, mô hình hoá biến phụ thuộc nhị phân. Về cơ bản, Logistic Regression là tìm mối quan hệ giữa các đặc trưng và xác suất của kết quả cụ thể (0 hoặc 1, yes hoặc no,..).

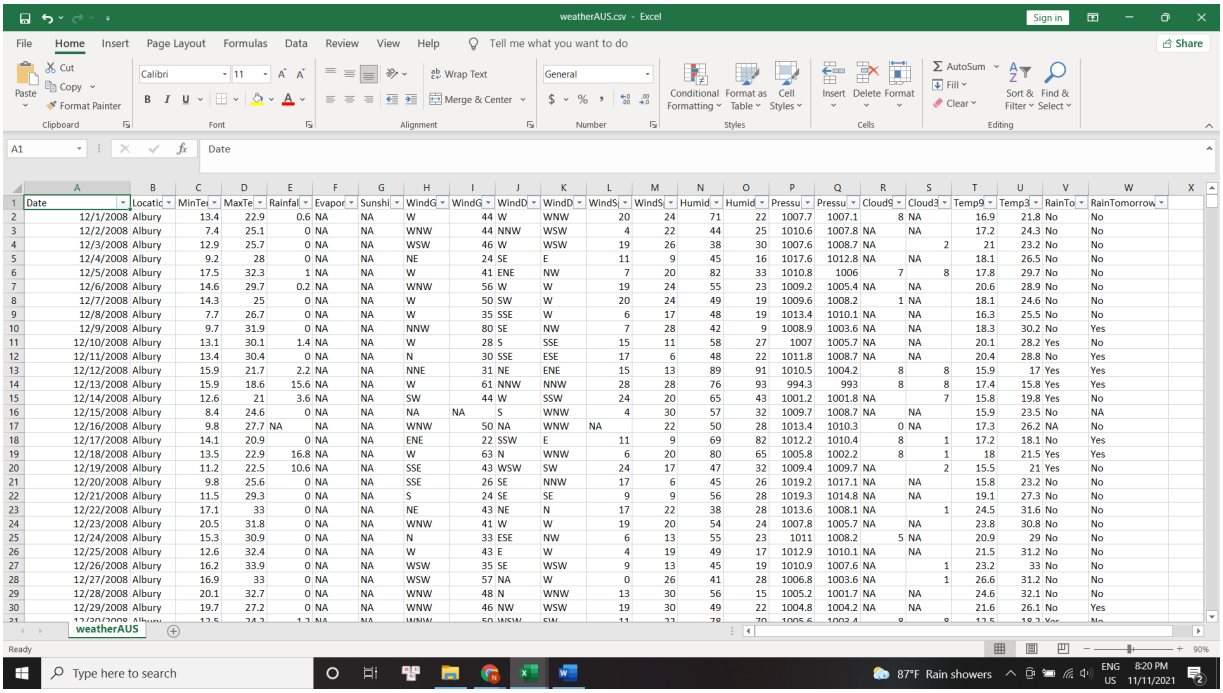
1.3 Mục tiêu đề tài

Dựa vào data weatherAUS.csv được được thu thập trong thực tế từ kaggle.com trong Rain in Australia để dự đoán dự đoán xác suất ngày tiếp theo ở Australia có mưa không.

1.4 Phạm vi đề tài

Nước Úc là một quốc gia bao gồm một đảo lớn là Tasmania, một tiểu bang của Úc và nhiều hòn đảo nhỏ ở Thái Bình Dương và Ấn Độ Dương. Với diện tích đất gần 7,7 triệu km2 và là quốc gia lớn thứ sáu về diện tích trên thế giới.

Dữ liệu có liên quan trực tiếp đến dự báo thời tiết của một quốc gia. Dữ liệu weatherAUS.csv được nghiên cứu bao gồm: Nhiệt độ, lượng mưa, độ bốc hơi, độ ẩm, áp xuất, hướng gió, vận tốc gió và mật độ mây được thu thập trong 10 năm (từ 1/12/2008 đến 25/6/2017) ở 49 trạm khí tượng của nước Úc.



Hình 2: Dataset thời tiết ở Úc

Link dataset: https://www.kaggle.com/jsphyg/weather-dataset-rattle-package

1.5 Phương pháp nghiên cứu

Tiến hành xử lý missing value.

Tiến hành phân tích data set theo phương pháp hold-out bằng cách chia tập dữ liệu thành 2 phần data train và test.

Áp dụng lý thuyết và thực thi đề tài bằng thuật toán hồi quy Logistic, thuật toán Machine Learning rừng ngẫu nhiên (Random Forest Machine Learning), thuật toán Naive Bayes và thuật toán XGBoost.

Trực quan hóa dữ liệu, đánh giá đo lường và cho ra kết quả.

1.6 Ý nghĩa khoa học và thực tiễn

Dự báo thời tiết ở Úc giúp cho cá nhân, tổ chức lên kế hoạch vào ngày tiếp theo từ đó phát triển những thuận lợi mà dự báo thời tiết mang đến.

Việc dự đoán trước thời tiết như vậy có ảnh hưởng lớn đến nhiều ngành nghề trong một quốc gia mà còn là nhu cầu thiết yếu hằng ngày của quốc gia đó.

Việc đơn giản giản khi dự báo thời tiết ngày mai có mưa không ảnh hưởng đến đời sống cá nhân như là phải chuẩn bị ô dù, áo mưa,....

Người làm nông nghiệp, dịch vụ, sản xuất...

Giúp cho những người làm khí tượng có thể dự đoán để đưa ra các cảnh báo và giải pháp cho người dân sống tốt hơn.

CHƯƠNG 2 - TỔNG QUAN GIẢI THUẬT

2.1 Giới thiệu về hồi quy logistic (Logistic Regression)

Trong thực tế, hồi quy logistic dùng để dự đoán xác suất xảy ra của một vấn đề cụ thể. Trong dự báo thời tiết, hồi quy logistic không chỉ dự đoán một ngày trong tương lai có mưa hay không, mà còn dự đoán được xác suất xảy ra mưa. Tương tự, hồi quy logistic có thể được sử dụng để dự đoán khả năng bệnh nhân mắc một bệnh cụ thể với các triệu chứng nhất định, đó là lý do tại sao nó rất phổ biến trong lĩnh vực y học.

Hồi quy logistic là một kỹ thuật thống kê được giám sát để tìm xác suất của biến phụ thuộc (Các lớp có trong biến).

Hồi quy logistic sử dụng các hàm được gọi là hàm logit, giúp suy ra mối quan hệ giữa biến phụ thuộc và các biến độc lập bằng cách dự đoán xác suất hoặc cơ hội xảy ra.

Các hàm logistic (còn được gọi là hàm sigmoid) chuyển đổi xác suất thành các giá trị nhị phân có thể được sử dụng thêm cho các dự đoán.

Logistic regression được áp dụng trong bài toán phân loại nhị phân (Binary classification) tức ta sẽ có hai output, hoặc có thể gọi là hai nhãn (ví dụ như 0 và 1 hoặc Yes or No).

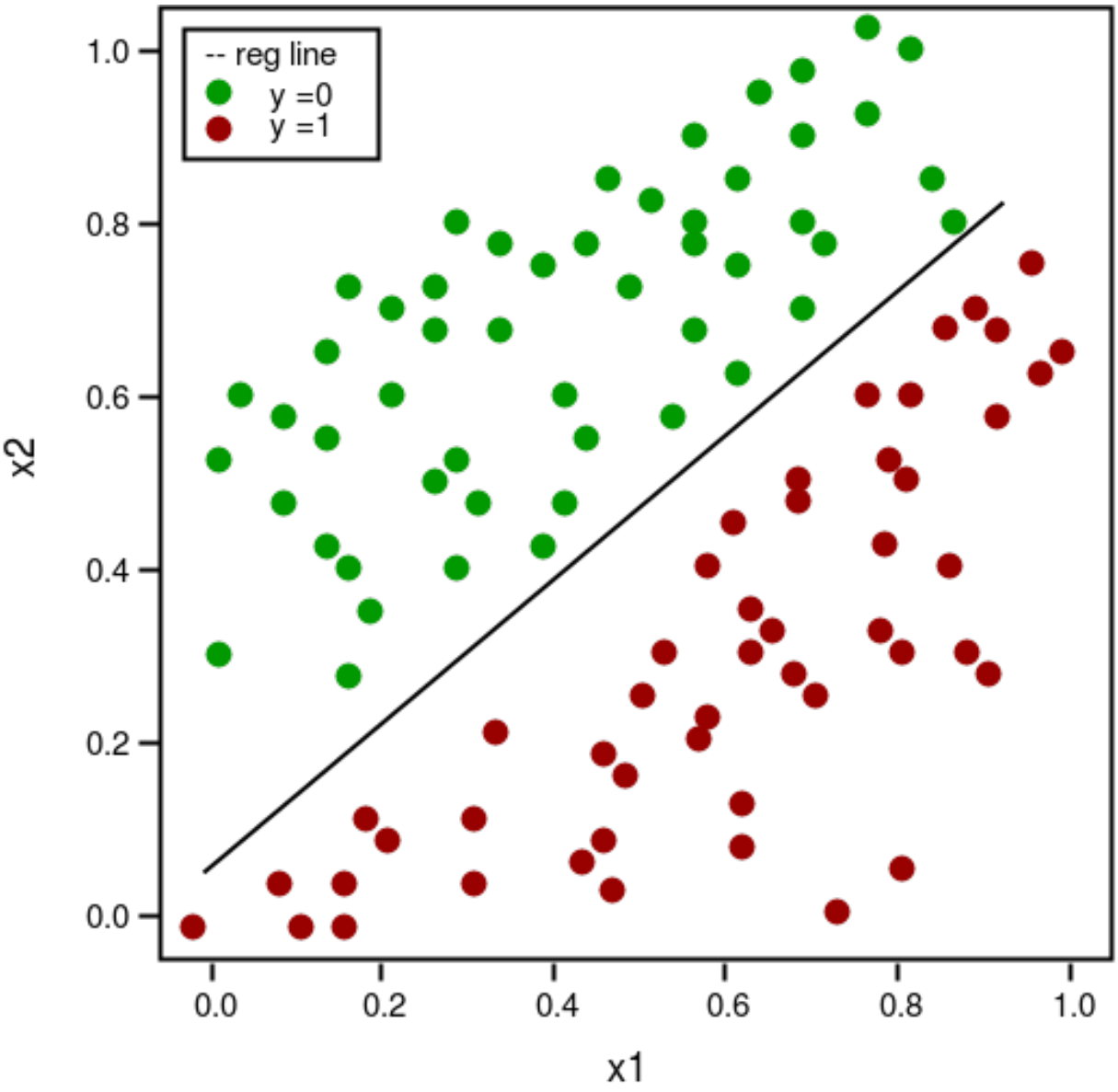
2.1.1 Trong thống kê

Hồi quy logistic là một phương pháp phân tích thống kê được sử dụng để dự đoán giá trị dữ liệu dựa trên các quan sát trước đó của tập dữ liệu.

Mục đích của hồi quy logistic là ước tính xác suất của các sự kiện, bao gồm xác định mối quan hệ giữa các tính năng từ đó dự đoán xác suất của các kết quả, nên đối với hồi quy logistic ta sẽ có:

Input: dữ liệu input (ta sẽ coi có hai nhãn là 0 và 1).

Output : Xác suất dữ liệu input rơi vào nhãn 0 hoặc nhãn 1.



Hình 3: Demo xác suất logistic
(Nguồn:https://benh.edu.vn/logistic-regression-la-gi/)

Ở hình 3 ta gọi các điểm màu xanh là nhãn 0 và các điểm màu đỏ là nhãn 1 đối với hồi quy logistic ta sẽ biết được với mỗi điểm thì xác xuất rơi vào nhãn 0 là bao nhiêu và xác suất rơi vào nhãn 1 là bao nhiêu, ta có thể thấy giữa hai màu xanh và màu đỏ có một đường thẳng để phân chia rất rõ ràng nhưng nếu các điểm dữ liệu mà không nằm sang hai bên mà nằm trộn lẫn nhiều vào nhau thì ta sẽ phân chia như nào, khi đó ta sẽ gọi tập dữ liệu có nhiều nhiễu và ta phải xử lí trước các nhiễu đó.

Mô hình hồi quy logistic là một mô hình thống kê được sử dụng rộng rãi, chủ yếu được sử dụng cho mục đích phân loại. Có nghĩa là với một tập hợp các quan sát, thuật toán hồi quy Logistic giúp chúng ta phân loại các quan sát này thành hai hoặc nhiều lớp rời rạc. Vì vậy, biến mục tiêu có bản chất rời rạc. Thuật toán hồi quy logistic hoạt động như sau:

Thuật toán hồi quy logistic hoạt động bằng cách thực hiện một phương trình tuyến tính với các biến độc lập hoặc giải thích để dự đoán giá trị phản hồi.

Phân tích hồi qui logistic là một kỹ thuật thống kê để xem xét mối liên hệ giữa biến độc lập (biến số hoặc biến phân loại) với biến phụ thuộc là biến nhị phân. Trong hồi qui tuyến tính đơn, biến độc lập x và phụ thuộc y là biến số liên tục liên hệ qua phương trình:

$$y = \alpha + \beta x + \varepsilon$$

Trong hồi qui logistic sử dujgn cho dự báo thời tiết, biến phụ thuộc y chỉ có 2 trạng thái 1 (có mưa) và 0 (không mưa). Muốn đổi ra biến số liên tục người ta tính xác suất của 2 trạng thái này. Nếu gọi p là xác suất để một biến cố xảy ra (có mưa), thì 1-p là xác suất để biến cố không xảy ra (không mưa). Phương trình hồi qui logistic phát biểu:

$$\text{Log}\left(\frac{p}{1-p}\right) = \alpha + \beta x + \varepsilon$$

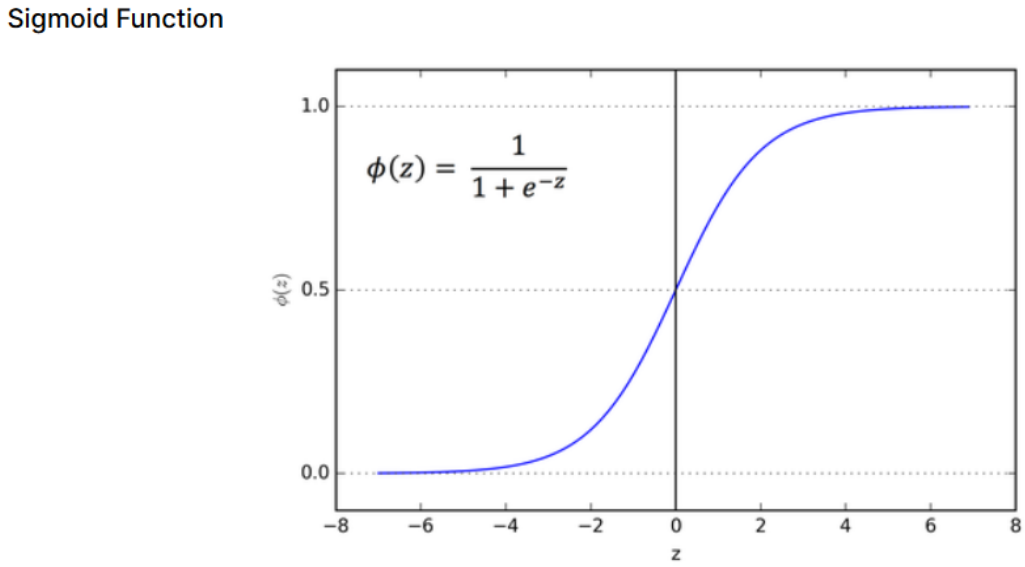
Từ phương trình này, ta có thể tính xác suất dự đoán có mưa theo trị số của x.

$$\frac{p}{1-p} = e^{\alpha + \beta x}$$

$$p = \frac{e^{\alpha + \beta x}}{1 + e^{\alpha + \beta x}}$$

2.1.2 Trong học máy

Hàm sigmoid được sử dụng để ánh xạ các dự đoán với xác suất. Hàm sigmoid có đường cong hình chữ S. Nó còn được gọi là đường cong sigmoid. Hàm Sigmoid là một trường hợp đặc biệt của hàm Logistic. Nó được đưa ra bởi công thức toán học và được biểu diễn như hình sau:



Hình 4: Đồ thị Sigmoid Function
(Nguồn:https://viblo.asia/p/logistic-regression-bai-toan-co-ban-trong-machine-learning-924lJ4)

Hàm sigmoid trả về giá trị xác suất từ 0 đến 1. Giá trị xác suất này sau đó được ánh xạ tới một lớp rời rạc là "0" hoặc "1". Để ánh xạ giá trị xác suất này tới một lớp rời rạc (đạt / không đạt, có / không, đúng / sai).

2.1.3 Phân loại Logistic Regression

Mô hình hồi quy logistic có thể được phân loại thành ba nhóm dựa trên các loại biến mục tiêu. Ba nhóm này được mô tả dưới đây:

Hồi quy logistic nhị phân:biến phụ thuộc chỉ có hai 2 kết quả / lớp có thể có.

Ví dụ: Nam hoặc Nữ, 0 hoặc 1, ác tính hoặc lành tính, đạt hoặc không đạt, được chấp nhận hoặc không được chấp nhận, có hoặc không, tốt hoặc xấu, đúng hoặc sai, spam hoặc không spam.

Hồi quy logistic đa thức: xử lý các trường hợp khi biến phụ thuộc chỉ có hai hoặc 3 kết quả / lớp có thể có trở lên mà không cần sắp xếp thứ tự.

Ví dụ: Dự đoán chất lượng thực phẩm. (Tốt, Tuyệt vời và Xấu), việc hình ảnh X-quang lồng ngực làm các đặc điểm cho biết về một trong ba kết quả có thể xảy ra (Không có bệnh, Viêm phổi do Vi rút, COVID-19).

Hồi quy Logistic thứ tự: biến phụ thuộc chỉ có hai hoặc nhiều hơn 3 kết quả / lớp có thể có với thứ tự.

Ví dụ: Xếp hạng sao từ 1 đến 5, kết quả học tập của học sinh có thể được phân loại là kém, trung bình, tốt và xuất sắc.

2.1.4 Ưu điểm khi sử dụng Logistic Regression

Logistic Regression là một trong những thuật toán học máy đơn giản nhất và dễ thực hiện nhưng mang lại hiệu quả lớn cũng như mô hình với thuật toán này không yêu cầu khả năng tính toán cao.

Các tham số dự đoán đưa ra suy luận về tầm quan trọng của từng tính năng. Hướng liên kết tức là tích cực hoặc tiêu cực cũng được đưa ra.

Thuật toán này cho phép các mô hình được cập nhật dễ dàng để phản ánh dữ liệu mới. Việc cập nhật có thể được thực hiện bằng cách sử dụng descent gradient ngẫu nhiên.

Logistic Regression đưa ra các xác suất được hiệu chỉnh tốt cùng với kết quả phân loại.

Trong một tập dữ liệu chồi quy logistic ít bị sai lệch hơn.

Hồi quy logistic đôi khi được sử dụng làm mô hình chuẩn để đo lường hiệu suất, vì nó tương đối nhanh chóng và dễ thực hiện.

Logistic Regression tỏ ra rất hiệu quả khi tập dữ liệu có các tính năng có thể phân tách tuyến tính.

2.1.5 Nhược điểm khi sử dụng Logistic Regression

Các vấn đề phi tuyến tính không thể được giải quyết bằng hồi quy logistic.

Không thể xử lý các vấn đề phức tạp bằng cách sử dụng hồi quy logistic và không dự đoán được kết quả liên tục.

2.1.6 Ứng dụng của Logistic Regression

Trong dự báo thời tiết, hồi quy logistic không chỉ dự đoán một ngày trong tương lai có mưa hay không, mà còn dự đoán được xác suất xảy ra mưa.

Trong ngành Giáo dục, hồi quy logistic có thể được sử dụng để dự đoán: Việc một học sinh có được nhận vào một chương trình đại học hay không dựa trên điểm thi và nhiều yếu tố khác. Trong các nền tảng E-learning để xem liệu học sinh có hoàn thành khóa học đúng hạn hay không dựa trên hoạt động trong quá khứ và các số liệu thống kê khác có liên quan đến vấn đề.

Trong lĩnh vực kinh doanh, hồi quy logistic có các ứng dụng dự đoán liệu giao dịch thẻ tín dụng của người dùng có gian lận hay không.

Ngành y tế cũng được hưởng lợi từ hồi quy logistic thông qua các cách sử dụng sau:dự đoán một người có mắc bệnh hay không dựa trên các giá trị thu được từ các báo cáo xét nghiệm hoặc các yếu tố khác nói chung. Một ứng dụng rất sáng tạo của Học máy đang được các nhà nghiên cứu sử dụng là dự đoán một người có COVID-19 hay không bằng cách sử dụng hình ảnh X-quang Ngực.

Các ứng dụng khác: Hồi quy logistic tìm thấy các ứng dụng của nó trong tất cả các lĩnh vực chính. Ngoài ra, một số ứng dụng thú vị của nó là: Phân loại email - Thư rác hay không phải thư rác. Phân tích cảm xúc - Một người đang buồn hay vui dựa trên một tin nhắn văn bản. Phát hiện và phân loại đối tượng - Phân loại hình ảnh thành hình ảnh mèo hoặc hình ảnh con chó.

Với dataset, tôi sẽ tạo mô hình hồi quy logistic sử dụng tập dữ liệu Rain in Australia để dự đoán liệu ngày mai trời có mưa hay không, sử dụng hồi quy logistic nhị phân. Mục đích của chúng tôi là tạo ra một mô hình để dự đoán giá trị trong cột RainTomorrow.

2.2 Giới thiệu về thuật toán Machine Learning rừng ngẫu nhiên (Random Forest Machine Learning)

Random Forest là một Machine Learning được sử dụng để giải quyết các vấn đề hồi quy và phân loại. Nó sử dụng phương pháp học tập hợp, là một kỹ thuật kết hợp nhiều bộ phân loại để đưa ra giải pháp cho các vấn đề phức tạp.

Random là ngẫu nhiên, Forest là rừng, nên ở thuật toán Random Forest mình sẽ xây dựng nhiều cây quyết định bằng thuật toán Decision Tree, tuy nhiên mỗi cây quyết định sẽ khác nhau). Sau đó kết quả dự đoán được tổng hợp từ các cây quyết định.

Random Forest là một tập hợp của các Decision Tree, mà mỗi cây được chọn theo một thuật toán dựa vào ngẫu nhiên.

Mỗi cây quyết định đều có những yếu tố ngẫu nhiên: Lấy ngẫu nhiên dữ liệu để xây dựng cây quyết định, lấy ngẫu nhiên các thuộc tính để xây dựng cây quyết định.

Decision Tree là tên đại diện cho một nhóm thuật toán phát triển dựa trên Cây quyết định. Ở đó, mỗi Node của cây sẽ là các thuộc tính, và các nhánh là giá trị lựa chọn của thuộc tính đó. Bằng cách đi theo các giá trị thuộc tính trên cây. Cây quyết định sẽ cho ta biết giá trị dự đoán. Nhóm thuật toán cây quyết định có một điểm mạnh đó là có thể sử dụng cho cả bài toán Phân loại (Classification) và Hồi quy (Regression).

Giả sử chúng ta muốn đi du lịch tại một địa điểm nào sắp tới, chúng ta 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. Chúng ta liền đi hỏi thêm một vài người nữa, và tổng hợp lại để cho ra quyết định đi địa điểm đó hay không.

Nếu coi mỗi ý kiến của những người góp ý là một cây quyết định, thì chúng ta đã có hình dung được thuật toán Random Forest.

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.

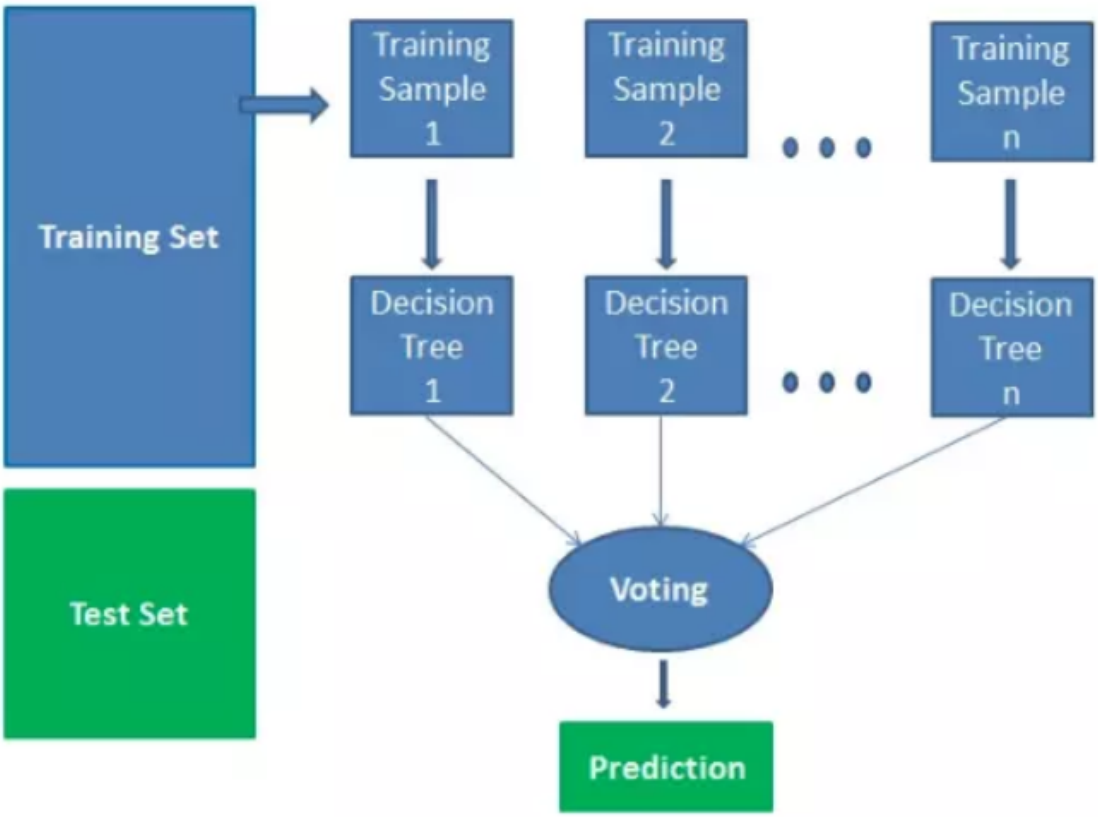
2.2.1 Cách thức hoạt động thuật toán Machine Learning rừng ngẫu nhiên

Chọn các mẫu ngẫu nhiên từ tập dữ liệu đã cho.

Thiết lập cây quyết định cho từng mẫu và nhận kết quả dự đoán từ mỗi cây quyết định.

Hãy bỏ phiếu cho mỗi kết quả dự đoán.

Chọn kết quả được dự đoán nhiều nhất là dự đoán cuối cùng.



Hình 5: Sơ đồ hoạt động Random Forest
(Nguồn:https://viblo.asia/p/phan-lop-bang-random-forests-trong-python-djeZ1D2QKWz)

2.2.2 Ưu điểm thuật toán Machine Learning rừng ngẫu nhiên

Áp dụng cho cả bài toán Phân loại (Classification) và Hồi quy (Regression).

Random forests cũng có thể xử lý các giá trị còn thiếu.

Xử lý tập dữ liệu lớn với kích thước lớn.

Có thể tạo mô hình cho các giá trị phân loại.

Nó không bị vấn đề overfitting.

2.2.3 Nhược điểm thuật toán Machine Learning rừng ngẫu nhiên

Random forests tạo ra dự đoán bởi chậm vì nó có nhiều cây quyết định.

Random forests tốn thời gian nhiều hơn các thuật toán khác.

2.2.4 Ứng dụng thuật toán Machine Learning rừng ngẫu nhiên

Ngân hàng: Rừng ngẫu nhiên được sử dụng trong ngân hàng để dự đoán mức độ tín nhiệm của người đi vay. Điều này giúp tổ chức cho vay đưa ra quyết định chính xác về việc có cho khách hàng vay hay không. Các ngân hàng cũng sử dụng thuật toán rừng ngẫu nhiên để phát hiện những kẻ gian lận.

Chăm sóc sức khỏe: Các chuyên gia y tế sử dụng hệ thống rừng ngẫu nhiên để chẩn đoán bệnh nhân. Bệnh nhân được chẩn đoán bằng cách đánh giá tiền sử bệnh trước đây của họ. Hồ sơ y tế trước đây được xem xét để thiết lập liều lượng phù hợp cho bệnh nhân.

Thị trường chứng khoán: Các nhà phân tích tài chính sử dụng nó để xác định thị trường tiềm năng cho cổ phiếu. Nó cũng cho phép họ xác định hành vi của cổ phiếu.

Thương mại điện tử: Thông qua các thuật toán rừng mưa, các nhà cung cấp thương mại điện tử có thể dự đoán mức độ yêu thích của khách hàng dựa trên hành vi tiêu dùng trong quá khứ.

Trong dự báo thời tiết, dự đoán một ngày trong tương lai có mưa hay không, mà còn dự đoán được xác suất xảy ra mưa.

2.3 Giới thiệu về thuật toán Naive Bayes

Naive Bayes Classification (NBC) là một thuật toán dựa trên định lý Bayes về lý thuyết xác suất để đưa ra các phán đoán cũng như phân loại dữ liệu dựa trên các dữ liệu được quan sát và thống kê. Naive Bayes là một trong những thuật toán được ứng dụng rất nhiều trong các lĩnh vực Machine learning dùng để đưa các dự đoán chính xác nhất dự trên một tập dữ liệu đã được thu thập, vì nó khá dễ hiểu và độ chính xác cao.

2.3.1 Định lý Bayes

Định lý Bayes cho phép tính xác suất xảy ra của một sự kiện ngẫu nhiên A khi biết sự kiện liên quan B đã xảy ra. Xác suất này được ký hiệu là P(A|B), và đọc là "xác suất của A nếu có B". Đại lượng này được gọi xác suất có điều kiện hay xác suất hậu nghiệm vì nó được rút ra từ giá trị được cho của B hoặc phụ thuộc vào giá trị đó.

Theo định lý Bayes, xác suất xảy ra A khi biết B sẽ phụ thuộc vào 3 yếu tố:

Xác suất xảy ra A của riêng nó, không quan tâm đến B. Kí hiệu là P(A) và đọc là xác suất của A. Theo nghĩa rằng nó không quan tâm đến bất kỳ thông tin nào về B.

Xác suất xảy ra B của riêng nó, không quan tâm đến A. Kí hiệu là P(B) và đọc là "xác suất của B". Đại lượng này còn gọi là hằng số chuẩn hóa (normalising constant), vì nó luôn giống nhau, không phụ thuộc vào sự kiện A.

Xác suất xảy ra B khi biết A xảy ra. Kí hiệu là P(B|A) và đọc là "xác suất của B nếu có A". Đại lượng này gọi là khả năng (likelihood) xảy ra B khi biết A đã xảy ra.

Gọi A, B là hai biến cố

Với P(B) > 0:

$$P(A \mid B) = \frac{P(AB)}{P(B)}$$

Suy ra:

$$P(AB) = P(A \mid B)P(B) = P(B \mid A)P(A)$$

Công thức Bayes:

$$P(B \mid A) = \frac{P(AB)}{P(A)} = \frac{P(A \mid B)P(B)}{P(A)} = \frac{P(A \mid B)P(B)}{P(AB) + P(A\overline{B})}$$

$$= \frac{P(A \mid B)P(B)}{P(AB) + P(A\overline{B})} = \frac{P(A \mid B)P(B)}{P(A \mid B)P(B) + P(A \mid \overline{B})P(\overline{B})}$$

Hình 6: Công thức Bayes

Nguồn: https://viblo.asia/p/thuat-toan-phan-lop-naive-bayes-924lJWPm5PM

Trong đó: P(A|B) là xác suất xảy ra của một sự kiện ngẫu nhiên A khi biết sự kiện liên quan B đã xảy ra. P(B|A) là xác suất xảy ra B khi biết A xảy ra. P(A) là xác suất sảy ra của riêng A mà không quan tâm đến B. P(B) là xác suất xảy ra của riêng B mà không quan tâm đến A.

Công thức Bayes tổng quát

Với P(A) > 0 và $\{B_1, B_2, ..., B_n\}$ là một hệ đầy đủ các biến cố:

- Tổng xác suất của hệ bằng 1:

$$\sum_{k=1}^{n} P(B_k) = 1$$

- Từng đôi một xung khắc:

$$P(B_i \cap B_j) = 0$$

Khi đó ta có:

$$P(B_k \mid A) = \frac{P(A \mid B_k)P(B_k)}{P(A)}$$

$$= \frac{P(A \mid B_k)P(B_k)}{\sum_{i=1}^{n} P(A \mid B_i)P(B_i)}$$

Hình 7: Công thức tổng quát

Nguồn: https://viblo.asia/p/thuat-toan-phan-lop-naive-bayes-924lJWPm5PM

2.3.2 Các loại phân phối dữ liệu Naive Bayes

Phân loại đầu tiên là Gaussian Naive Bayes được sử dụng chủ yếu trong loại dữ liệu mà các thành phần là các biến liên tục. Gaussian (hoặc phân phối chuẩn) là phéptính giá trị trung bình và độ lệch chuẩn từ dữ liệu traiing set. Mô hình Gaussian Naive Bayes từ dữ liệu:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$

Hình 8: Công thức Gaussian Naive Bayes
Nguồn: https://viblo.asia/p/mo-hinh-phan-lop-naive-bayes-vyDZO0A7lwj

Hoặc ta có thể biểu diễn công thức:

pdf (x, trung bình, sd) = (1 / (sqrt (2 * PI) * sd)) * exp (- ((x-mean ^{2})/(2∗sd^2)))

Trong đó pdf (x) là Gaussian PDF, sqrt () là căn bậc hai, giá trị trung bình và sd là giá trị trung bình và độ lệch chuẩn được tính ở trên, PI là hằng số, exp () là hằng số e hoặc Euler được nâng lên thành công suất và x là giá trị đầu vào cho biến đầu vào.

Phân loại thức hai là Bernoulli Naive Bayes, mô hình này được áp dụng cho các loại dữ liệu mà mỗi thành phần là một giá trị nhị phân bằng 0 hoặc 1. Ví dụ: cũng với loại văn bản nhưng thay vì đếm tổng số lần xuất hiện của 1 từ trong văn bản, ta chỉ cần quan tâm từ đó có xuất hiện hay không.

Khi đó p(xi|c) được tính bằng:

$$p(x_i|c) = p(i|c)^{x_i}(1 - p(i|c))^{1-x_i}$$

Hình 9: Công thức Bernoulli Naive Bayes
Nguồn: https://www.dataisg.org/tutorial/machine-learning/naive-bayes-classifier

Với p(i|c) là xác xuất từ thứ i xuất hiện trong văn bản của lớp c.

Phân loại thứ ba là Multinomial Naive Bayes, mô hình này chủ yếu được sử dụng trong phân loại văn bản. Đặc trưng đầu vào ở đây chính là tần suất xuất hiện của từ trong văn bản đó.

Khi đó, p(xi|c) tỉ lệ với tần suất từ thứ i xuất hiện trong văn bản của class c. Giá trị này được tính bằng:

$$\lambda_{ci} = p(x_i|c) = \frac{N_{ci}}{N_c}$$

Hình 10: Công thức Multinomial Naive Bayes
Nguồn: https://www.dataisg.org/tutorial/machine-learning/naive-bayes-classifier

Với: N(ci) là số lần từ thứ i xuất hiện trong văn bản của lớp c. Được tính bằng tổng của tất cả các thành phần thứ i của vectơ đặc trưng ứng với lớp c. Nc là tổng số từ xuất hiện trong lớp c, bằng tổng độ dài của toàn bộ văn bản thuộc lớp c.

2.3.3 Ưu điểm

Dễ sử dụng và nhanh khi cần dự đoán nhãn của dữ liệu test. Thực hiện khá tốt trong multiclass prediction.

Có thể hoạt động với các vectơ đặc trưng mà một phần là liên tục (sử dụng Gaussian Naive Bayes), phần còn lại ở dạng rời rạc (sử dụng Multinomial hoặc Bernoulli).

Khi giả định rằng các đặc trưng của dữ liệu là độc lập với nhau thì Naive Bayes chạy tốt hơn các thuật toán khác như logistic regression khi có ít dữ liệu đào tạo.

Trực quan khi bạn hiểu khái niệm

Rất dễ triển khai và hoạt động tốt trong dự đoán đa kính

Nó hoạt động tốt với các biến đầu vào phân loại

2.3.4 Nhược điểm

Độ chính xác của Naive Bayes nếu so với các thuật toán khác thì không được cao.

Bất khả thi khi các đặc trưng của dữ liệu là độc lập với nhau.

2.3.5 Ứng dụng

Dự đoán theo thời gian thực: Vì Naive Bayes nhanh và dựa trên số liệu thống kê của Bayes nên nó hoạt động tốt trong việc đưa ra dự đoán trong thời gian thực.

Trên thực tế, rất nhiều mô hình thời gian thực hoặc mô hình trực tuyến phổ biến dựa trên số liệu thống kê của Bayes.

Dự đoán đa kính (Multiclass prediction): Như đã nói trước đây, Naive Bayes hoạt động tốt khi có nhiều hơn hai lớp cho biến đầu ra.

Phân loại văn bản (Text classification/ Spam Filtering/ Sentiment Analysis): Phân loại văn bản cũng bao gồm các ứng dụng phụ như lọc thư rác và phân tích tình cảm. Vì Naive Bayes hoạt động tốt nhất với các biến rời rạc, nó có xu hướng hoạt động tốt trong các ứng dụng này.

Hệ thống đề xuất (Recommendation System): Naive Bayes thường được sử dụng cùng với các thuật toán khác như lọc cộng tác để xây dựng các hệ thống đề xuất như phần đề xuất của Netflix hoặc các sản phẩm được đề xuất của Amazon hoặc các bài hát được đề xuất của Spotify.

Với dataset, tôi sẽ tạo mô hình Bernoulli Naive Bayes và Gaussian Naive Bayes sử dụng tập dữ liệu Rain in Australia để dự đoán liệu ngày mai trời có mưa hay không. Mục đích của chúng tôi là tạo ra một mô hình để dự đoán giá trị trong cột RainTomorrow.

2.4 Giới thiệu về thuật toán XGBoost

Trong các vấn đề dự đoán liên quan đến dữ liệu phi cấu trúc (hình ảnh, văn bản, v.v.), mạng neural có xu hướng vượt trội hơn tất cả các thuật toán hoặc framework khác. Tuy nhiên, khi nói đến dữ liệu dạng bảng / dữ liệu có cấu trúc các thuật toán tree-based xử lý rất tốt.

Thuật toán XGBoost thuộc nhóm các thuật toán tree-based được phát triển như một dự án nghiên cứu tại Đại học Washington và là phiên bản cải tiến của Gradient Boosting.

XGBoost có thể được sử dụng để giải quyết được tất cả các vấn đề từ hồi quy (regression), phân loại (classification), ranking và giải quyết các vấn đề do người dùng tự định nghĩa.

Tốc độ xử lý: XGBoost thực hiện tinh toán song song nên tốc độ xử lý có thể tăng gấp 10 lần so với GBM. Ngoài ra, XGboost còn hỗ trợ tính toán trên Hadoop.

Overfitting: XGBoost áp dụng cơ chế Regularization nên hạn chế đáng kể hiệ tượng Overfitting (GBM không có regularization).

Sự linh hoạt: XGboost cho phép người dùng sử dụng hàm tối ưu và chỉ tiêu đánh giá của riêng họ, không hạn chế ở những hàm cung cấp sẵn.

Xử lý missing value: XGBoost bao gồm cơ chế tự động xử lý missing value bên trong nó. Vì thế, có thể bỏ qua bước này khi chuẩn bị dữ liệu cho XGBoost.

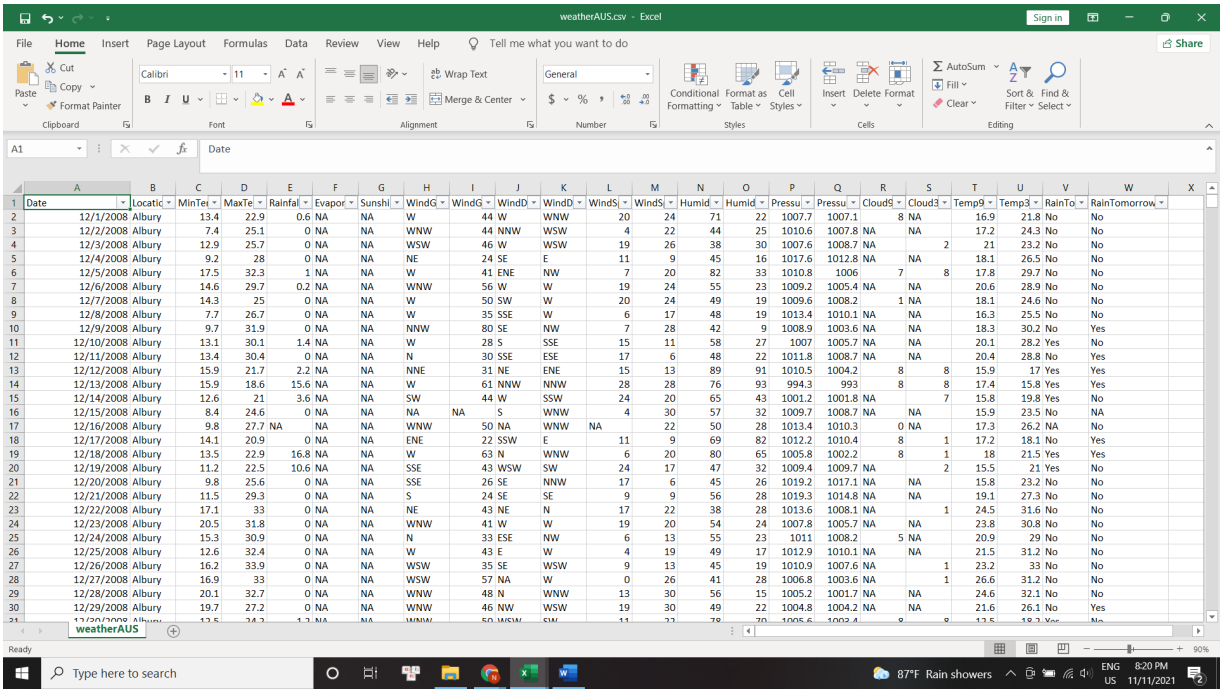
Tự động cắt tỉa: Tính năng tree pruning hộ trợ việc tự động bỏ qua những leaves, nodes không mang giá trị tích cực trong quá trình mở rộng tree.

Với dataset, tôi sẽ tạo mô hình XGBoost sử dụng tập dữ liệu Rain in Australia để dự đoán liệu ngày mai trời có mưa hay không. Loading dữ liệu, phân tách dữ liệu xử lý missing value và tạo train test. Mục đích của chúng tôi là tạo ra một mô hình để dự đoán giá trị trong cột RainTomorrow.

CHƯƠNG 3 - DỮ LIỆU THỰC NGHIỆM

3.1 Đặc tả dữ liệu thực nghiệm

Dữ liệu có liên quan trực tiếp đến dự báo thời tiết của một quốc gia. Dữ liệu weatherAUS.csv được nghiên cứu bao gồm: Nhiệt độ, lượng mưa, độ bốc hơi, độ ẩm, áp xuất, hướng gió, vận tốc gió và mật độ mây được thu thập trong 10 năm (từ 1/12/2008 đến 25/6/2017) ở 49 trạm khí tượng của nước Úc.



Hình 11: Dataset thời tiết ở Úc

Link dataset: https://www.kaggle.com/jsphyg/weather-dataset-rattle-package

Dữ liệu thực tế được ghi nhận tại Australia trong 10 năm liền có 145460 dòng với 23 thuộc tính bao gồm 2 thuộc tính RainToday, RainTomorrow.

Date: ngày tháng năm thu thập dòng dữ liệu (mm/dd/yyyy).

Location: địa điểm (name).

MinTemp: nhiệt độ tối thiểu (oC).

MaxTemp: nhiệt độ tối đa (oC).

Rainfall: lượng mưa được ghi nhận trong ngày (mm).

Evaporation: sự bốc hơi được ghi nhận từ 24 giờ đêm đến 9h sáng (mm).

Sunshine: số giờ nắng trong ngày (hour).

WindGustDir: hướng gió (N-E-W-S).

WindGustSpeed: vận tốc gió (km/h).

WindDir9am: hướng gió lúc 9 giờ sáng (N-E-W-S).

WindDir3pm: hướng gió lúc 3 giờ chiều (N-E-W-S).

WindSpeed9am: vận tốc gió lúc 9 giờ sáng (km/h).

WindSpeed3pm: vận tốc gió lúc 3 giờ chiều (km/h).

Humidity9am: độ ẩm không khí lúc 9 giờ sáng (g/m3).

Humidity3pm: độ ẩm không khí lúc 3 giờ chiều (g/m3).

Pressure9am: áp suất không khí lúc 9 giờ sáng (N/m2).

Pressure3pm: áp suất không khí lúc 3 giờ chiều (N/m2).

Cloud9am: mật độ mây theo các mức độ lúc 9 giờ sáng (0-9).

Cloud3pm: mật độ mây theo các mức độ lúc 3 giờ chiều (0-9).

Temp9am: nhiệt độ ghi nhận lúc 9 giờ sáng (oC).

Temp3pm: nhiệt độ ghi nhận lúc 3 giờ chiều (oC).

RainToday: thuộc tính ghi nhận mưa trong ngày (yes/no).

RainTomorrow: thuộc tính dự đoán mưa trong ngày tiếp theo (yes/no).

Với các dữ liệu bị thiếu trong dataset weatherAUS.csv sẽ được xử lý missing value như sau:

Bước 1: Loading dataset

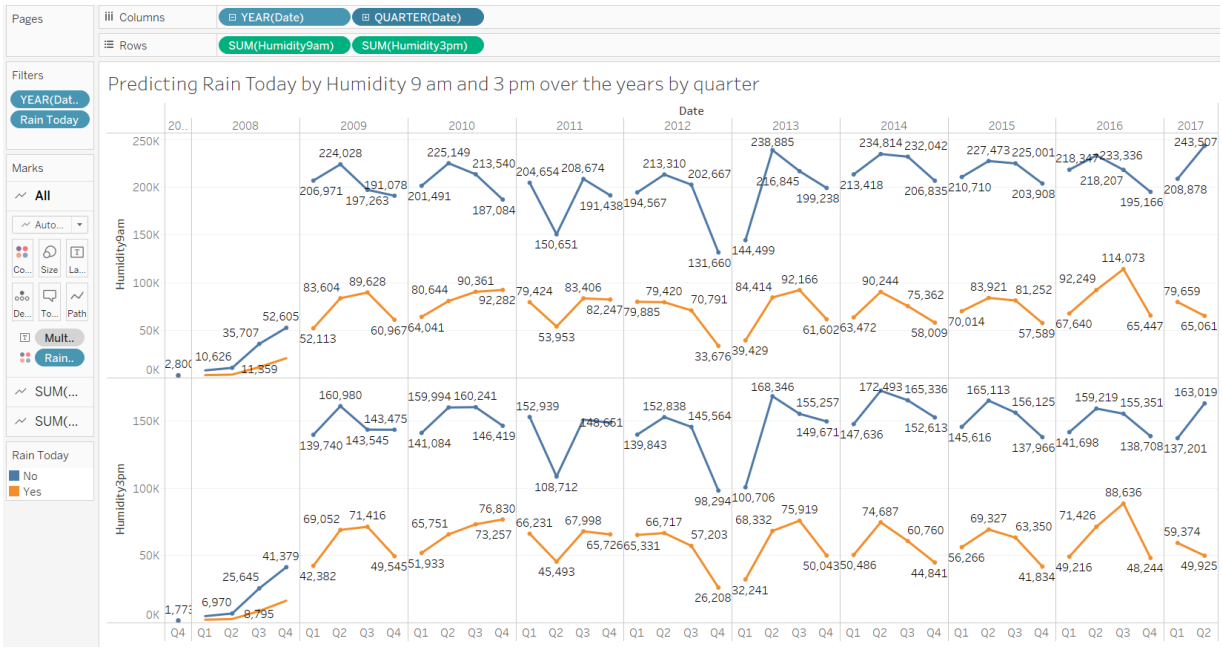
Bước 2: Phân tích các thuộc tính thành 2 loại (kiểu số và kiểu kí tự).

Bước 3: Có nhiều cách xử lý dữ liệu bị thiếu: Sử dụng trung vị hoặc giá trị trung bình, chọn giá trị ngẫu nhiên để đưa vào hoặc giá trị có tần suất xuất hiện nhiều nhất của mỗi thuộc tính

Ở trường hợp dataset này sử dụng điền các giá trị còn thiếu bằng giá trị có tần suất xuất hiện nhiều nhất của mỗi thuộc tính

Sử dụng hàm fillna() để tìm dữ liệu xuất hiện nhiều nhất lấp vào chỗ trống.

3.2 Thực nghiệm bằng đồ thị dùng phần mềm Tableau

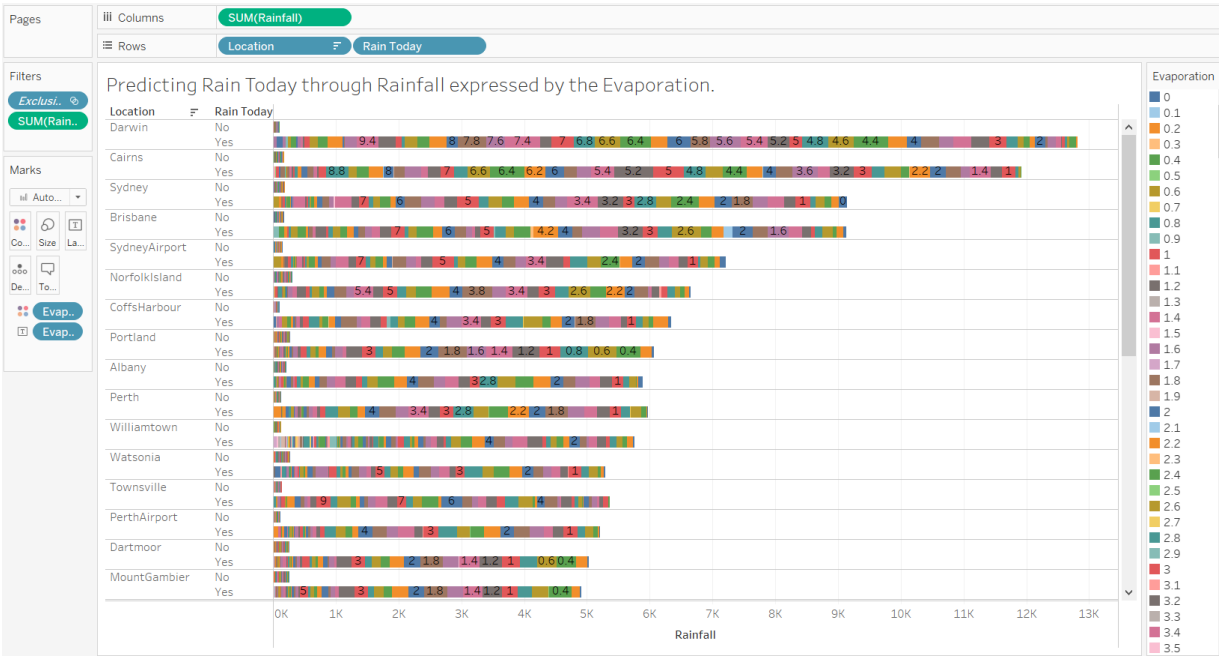


Hình 12: Biểu đồ dự đoán hôm nay trời mưa qua độ ẩm 9h sáng và 3h chiều theo từng quý (2007-2017)

Biểu diễn được sự khác biệt giữa độ ẩm lúc 9 giờ sáng và 3 giờ chiều cho thấy được tổng độ ẩm vào lúc 3 giờ ở nước Úc thường thấp hơn lúc 9 giờ sáng thấp nhất là 1773 g/m3 thấp hơn là 2800 g/m3 độ ẩm thấp nhất vào lúc 9 giờ sáng được ghi số liệu vào năm 2007 quý 4. Và tổng độ ẩm cao nhất 9 giờ sáng là 243507 g/m3 vào quý 2 năm 2017 với buổi chiều độ ẩm cao nhất được ghi nhận là vào quý 2 năm 2014 là 1724936 g/m3. Kết luận dù cho độ ẩm cao nhưng cũng không trời mưa thông qua dự đoán giá trị No, phân tích cho thấy độ ẩm No cao hơn phần giá trị Yes, cho nên hôm nay có mưa không phụ thuộc vào độ ẩm cao.

Hình 13: Biểu đồ cột thể hiện số tổng lượng mưa sẽ rơi qua dự đoán ngày mai có mưa hay không (2007-2017).

Tổng lượng mưa sẽ rơi thể hiện qua giá trị Yes từ năm 2008 cao hơn hẳn tổng lượng mưa sẽ không rơi qua giá trị No. Minh chứng từ năm 2008 giá trị Yes là 2671 mm cao hơn với 2470 mm giá trị No. Và tổng lượng mưa rơi cao nhất của Yes là 25980mm so với lượng mưa No trong năm đó chỉ 17 849 mm khẳng định năm đó sẽ trời đổ mưa nhiều hơn so các năm còn lại. Kết luận có thể năm 2007 có lượng mưa giá trị Yes thấp hơn lượng mưa so với gái trị No thì năm đó RainTomorrow có dự báo thời tiết trời đổ mưa ít hơn so với các năm khác.

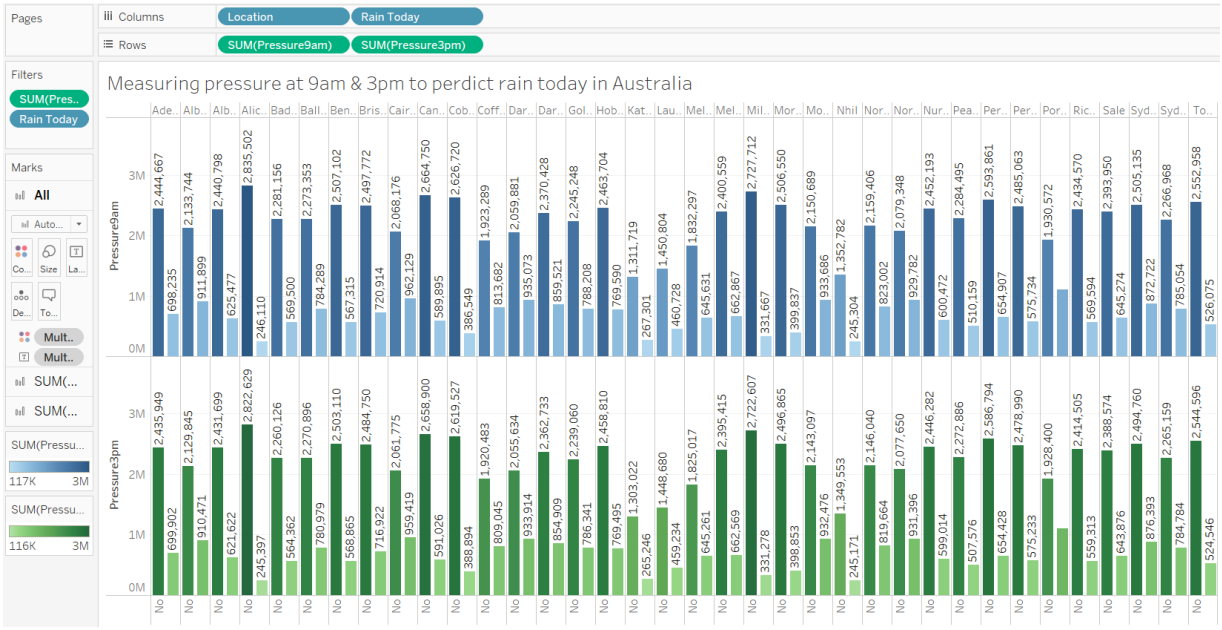


Hình 14: Biểu đồ cột ngang thể hiện dự đoán hôm nay trời sẽ mưa qua tổng lượng mưa sẽ rơi thể hiện theo thuộc tính Evaporation(sự bốc hơi)

Biểu đồ cho biết sự bốc hơi ở khắp nước Úc kèm theo lượng mưa được ghi nhận trong ngày có qua dự đoán hôm nay có mưa hay không(Yes/No), có bỏ qua một vài địa điểm có giá trị null về sự bốc hơi, nơi được ghi nhận số liệu sự bốc hơi của nước Úc tốt nhất là Darwin có chỉ số cụ thể rõ ra và lượng mưa trong ngày ghi nhận cũng cao nhất gần 13000mm và với độ bốc hơi được thể hiện liên tục từ 0 cho đến 8.8 mm. Kết luận rằng ngày có lượng mưa thấp cũng lại có giá trị dự đoán No, trời sẽ không mưa, hiển nhiên có lượng mưa cao và thể hiện được độ bốc hơi cao thì có khả năng trời sẽ mưa thể hiện qua giá trị Yes.

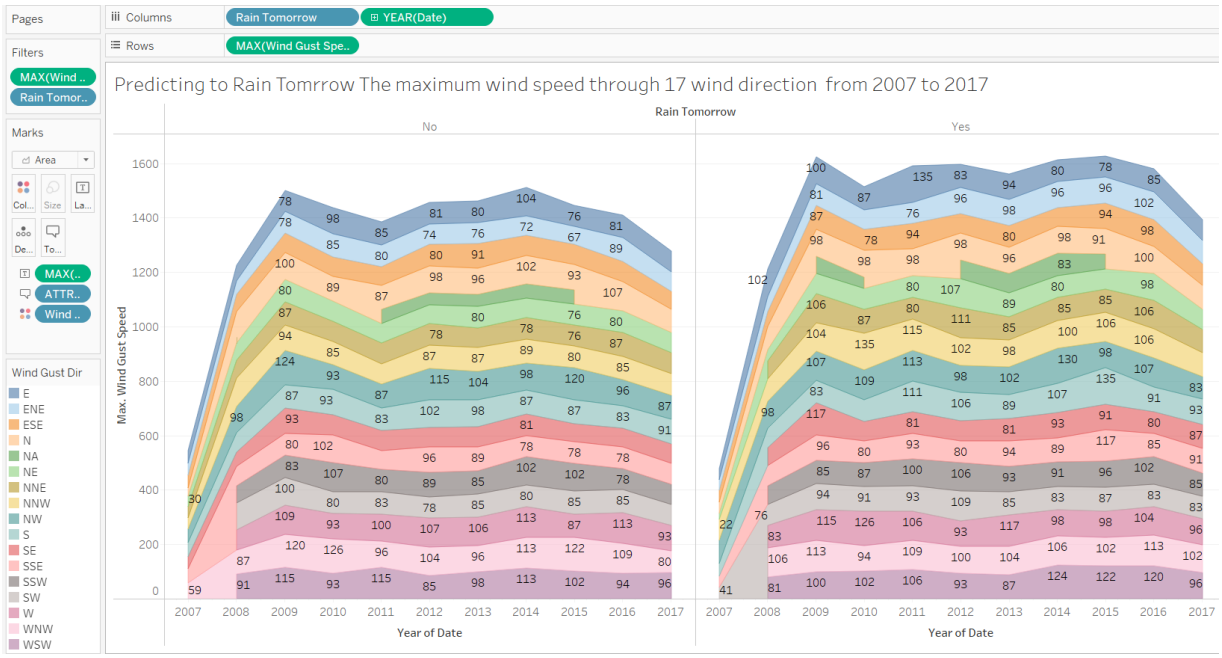
Hình 15: Biểu đồ cột ngang thể hiện số giờ nắng được ghi nhận trong ngày kèm theo tổng lượng mưa sẽ rơi ở khắp nơi nước Úc.

Cho ta thấy rằng Darwin là nơi ghi nhận tốt nhất về tổng lượng mưa cao nhất như khoảng 3192 mm, số giờ nắng ghi nhận được trong ngày dựa vào bảng màu bên phải thì tương ứng là 0 giờ. Và có xóa đi những nơi không ghi nhận được số giờ nắng có giá trị null, và nơi có tổng lượng mưa được ghi nhận trong ngày thấp là Cobar được ghi nhận với lượng mưa chưa tới 1000 mm thì có lẽ số ngày nắng ghi nhận được có khi lại cao hơn. Do nếu ghi nhận được số giờ nắng cao thì tổng lượng mưa hôm đó rới sẽ rất thấp vì chúng nó trái nghịch nhau.



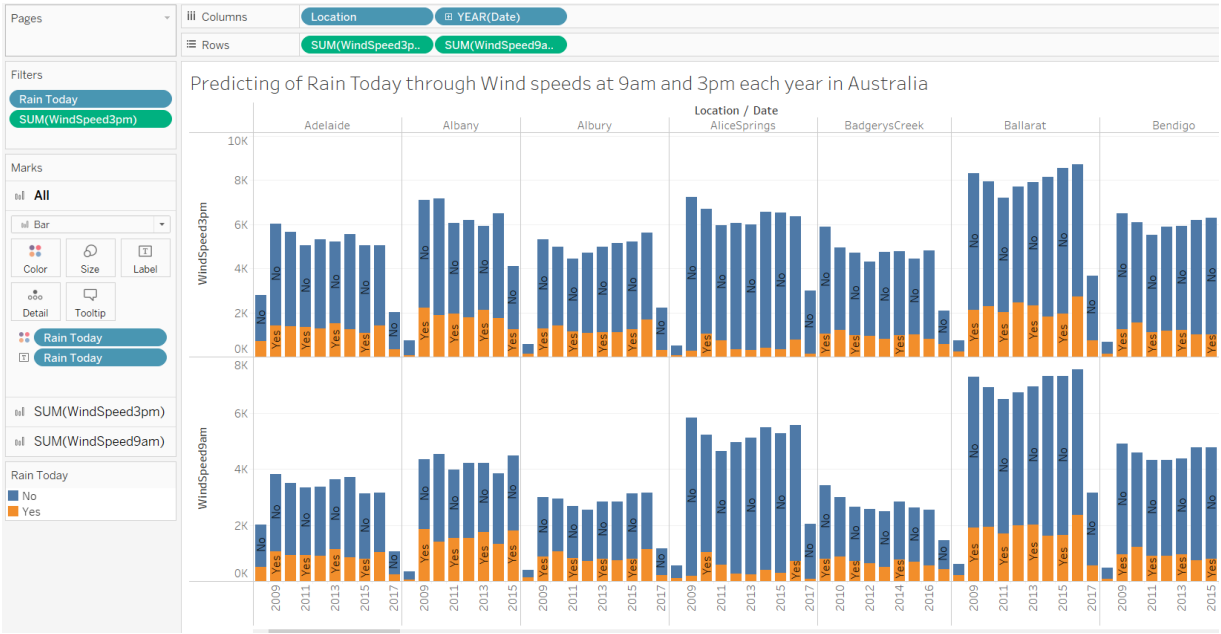
Hình 16: Biểu đồ cột hiển thị tổng áp suất không khí đo được lúc 9 giờ sáng và 3 giờ chiều ở nước Úc

Tổng áp suất không khí lúc 9 giờ sáng luôn lớn hơn tổng áp suất không khí lúc 3 giờ chiều. Và áp suất không khí cao thì ghi nhận mưa trong ngày là giá trị No, vậy nghĩa là ấp suất không khí thấp thì hôm đó ghi nhận được trời sẽ đổ mưa trong ngày.



Hình 17: Biểu đồ rời rạc dự đoán mưa vào ngày tiếp theo với vận tốc gió lúc 24 giờ lớn nhất qua các hướng gió (2007 – 2017)

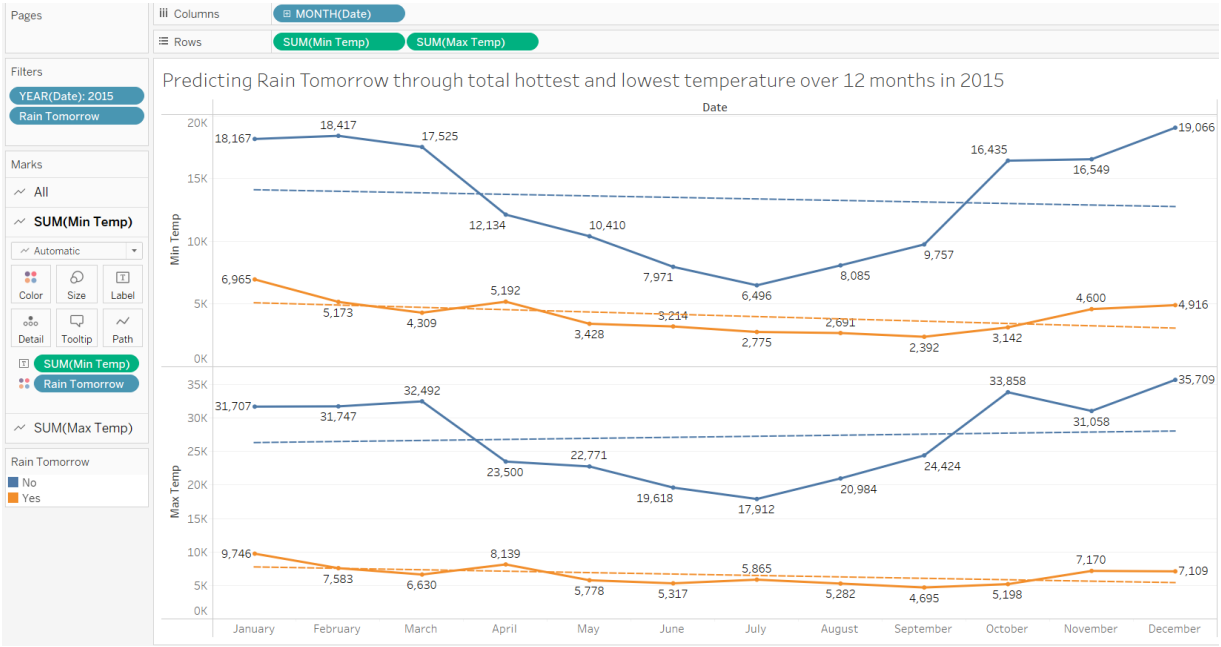
Vận tốc gió lớn nhất lên tới 135 km/h hay thổi qua hướng NNW là hướng Bắc Tây Bắc và S là hướng phía Nam cụ thể tuần tự từng đợt là vào năm 2010 và năm 2015 qua giá trị Yes. Qua đó ta thấy được dự đoán mai sẽ mưa thì vận tốc gió lúc 24 giờ cũng lớn hơn.



Hình 18: Biểu đồ cột ghi nhận mưa trong ngày qua tốc độ gió lúc 9 giờ sáng và 3 giờ chiều qua mỗi năm ở nước Úc

Hình 19: Biểu đồ bảng dự đoán mai những nơi ở nước Úc sẽ có mưa vào ngày 11 mỗi tháng từ năm 2007-2017

Qua đó ta thấy được nước Úc rất ít những ngày dự đoán mai có mưa chỉ chọn qua một ngày cụ thể của mỗi tháng trải qua 10 năm.



Hình 20: Dự đoán mưa trong ngày tiếp theo qua tổng nhiệt độ nóng nhất và thấp nhất qua 12 tháng trong năm 2015

Nhiệt độ cao thì sẽ ghi nhận mưa không có mưa vào ngày tiếp theo, và ngược lại.

CHƯƠNG 4 - THỰC NGHIỆM

4.1 Hiện thực giải thuật và thực nghiệm trên dữ liệu

Thêm thư viện

```
import numpy as np ## Đại số tuyến tính
import pandas as pd ## Xử lý dữ liệu
import matplotlib ## Thư viện sử dụng để vẽ các đồ thị trong Python
import seaborn as sns# Trực quan hóa dữ liệu thống kê
import matplotlib.pyplot as plt ## Vẽ đồ thị trực quan hóa dữ liệu
from sklearn.model_selection import train_test_split## Thư viện dùng để phân chia vùng dữ liệu
from scipy import stats##Các hàm và phân phối thống kê
from sklearn.linear_model import LogisticRegression ## hàm LogisticRegression
from collections import Counter##Hàm đếm
from sklearn.metrics import confusion_matrix##ma trận vuông mô hình hóa
from sklearn.metrics import accuracy_score
from xgboost import XGBClassifier ## thư viện hàm xgboost
from sklearn.naive_bayes import BernoulliNB ##Thư viện chứa hàm naive_bayes Bernoulli
from sklearn.naive_bayes import GaussianNB##Thư viện chứa hàm naive_bayes Gaussian
from sklearn.ensemble import RandomForestRegressor ##Thư viện chứa hàm RandomForest
from sklearn.svm import SVC
from sklearn import preprocessing
from scipy import stats
import warnings
warnings.filterwarnings("ignore")
```

Hình 21: Thêm thư viện vào

Đọc dữ liệu từ .csv và xem kích thước dữ liệu

```
[2]  data=pd.read_csv('weatherAUS.csv') # Loading the Data
```

Hình 22: Loading dữ liệu

In ra dữ liệu

```
data##In ra dữ liệu
```

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	WindSpeed9am	WindSpeed3pm	Humidity9am	Humidity3pm	P
0	12/1/2008	Albury	13.4	22.9	0.6	NaN	NaN	W	44.0	W	WNW	20.0	24.0	71.0	22.0	
1	12/2/2008	Albury	7.4	25.1	0.0	NaN	NaN	WNW	44.0	NNW	WSW	4.0	22.0	44.0	25.0	
2	12/3/2008	Albury	12.9	25.7	0.0	NaN	NaN	WSW	46.0	W	WSW	19.0	26.0	38.0	30.0	
3	12/4/2008	Albury	9.2	28.0	0.0	NaN	NaN	NE	24.0	SE	E	11.0	9.0	45.0	16.0	
4	12/5/2008	Albury	17.5	32.3	1.0	NaN	NaN	W	41.0	ENE	NW	7.0	20.0	82.0	33.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
161149	6/21/2018	Woomera	7.0	33.0	140.0	41.0	10.0	SE	66.0	4	E	104.0	7.0	3.0	96.0	
161150	6/22/2018	Woomera	-1.0	-3.0	220.0	32.0	8.0	SE	97.0	49	E	34.0	37.0	97.0	48.0	
161151	6/23/2018	Woomera	4.0	18.0	158.0	108.0	4.0	SE	129.0	5	E	70.0	27.0	41.0	7.0	
161152	6/24/2018	Woomera	23.0	20.0	172.0	128.0	9.0	SE	74.0	21	E	95.0	70.0	90.0	95.0	
161153	6/25/2018	Woomera	32.0	43.0	190.0	31.0	11.0	SE	89.0	78	E	84.0	40.0	70.0	20.0	

161154 rows × 23 columns

Hình 23: Hiển thị dữ liệu

Tổng giá trị dữ liệu và thuộc tính

```
data.shape##Tổng giá trị dữ liệu và thuộc tính
```

```
(161154, 23)
```

Hình 24: Tổng giá trị dữ liệu và thuộc tính

Tìm tất cả các giá trị kiểu số và kiểu kí tự trong thuộc tính

```
##Tìm tất cả các giá trị kiểu số và kiểu kí tự trong thuộc tính
categorical_col, contin_val=[],[]
for i in data.columns:
    if data[i].dtype == 'object':
        categorical_col.append(i)
    else:
        contin_val.append(i)
print(categorical_col)
print(contin_val)
```

```
['Date', 'Location', 'WindGustDir', 'WindDir9am', 'WindDir3pm', 'RainToday', 'RainTomorrow']
['MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation', 'Sunshine', 'WindGustSpeed', 'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am', 'Humidity3pm', 'Pressure9am', 'Pressure3pm', 'Cloud9am', 'Cloud3pm', 'Temp9am', 'Temp3pm']
```

Hình 25: Tìm tất cả các giá trị kiểu số và kiểu kí tự trong thuộc tính

Số lượng các giá trị duy nhất trong cột (ví dụ cột RainTomorrow có 2 giá trị Yes No)

```
data.nunique()#số lượng các giá trị duy nhất trong cột (ví dụ cột RainTomorrow có 2 giá trị Yes No)
```

```
Date             3801
Location           49
MinTemp           392
MaxTemp           507
Rainfall          934
Evaporation       449
Sunshine          145
WindGustDir        16
WindGustSpeed     132
WindDir9am        104
WindDir3pm         16
WindSpeed9am      131
WindSpeed3pm       88
Humidity9am       101
Humidity3pm       101
Pressure9am       551
Pressure3pm       555
Cloud9am           10
Cloud3pm           10
Temp9am           444
Temp3pm           504
RainToday           2
RainTomorrow        2
dtype: int64
```

Hình 26: Số lượng các giá trị duy nhất trong cột

Kiểm tra các giá trị Null trong data

```
data.isnull().sum()# kiểm tra các giá trị Null trong data
```

```
Date                0
Location            0
MinTemp          1485
MaxTemp          1261
Rainfall         3261
Evaporation     62790
Sunshine        69835
WindGustDir     10354
WindGustSpeed   10263
WindDir9am      10566
WindDir3pm       4231
WindSpeed9am     1767
WindSpeed3pm     3062
Humidity9am      2654
Humidity3pm      4507
Pressure9am     15065
Pressure3pm     15028
Cloud9am        55888
Cloud3pm        59358
Temp9am          1767
Temp3pm          3609
RainToday        3261
RainTomorrow     3267
dtype: int64
```

Hình 27: Kiểm tra các giá trị Null trong data

Thay đổi yes và No thành 1 và 0 trong một số cột

```
data['RainTomorrow'] = data['RainTomorrow'].map({'Yes': 1, 'No': 0})#Thay đổi yes và No thành 1 và 0
data['RainToday'] = data['RainToday'].map({'Yes': 1, 'No': 0})#Thay đổi yes và No thành 1 và 0
print(data.RainToday)
print(data.RainTomorrow)
```

```
0         0.0
1         0.0
2         0.0
3         0.0
4         0.0
         ...
161149    0.0
161150    0.0
161151    0.0
161152    0.0
161153    0.0
Name: RainToday, Length: 161154, dtype: float64
0         0.0
1         0.0
2         0.0
3         0.0
4         0.0
         ...
161149    0.0
161150    0.0
161151    0.0
161152    0.0
161153    0.0
Name: RainTomorrow, Length: 161154, dtype: float64
```

Hình 28: Thay đổi yes và No thành 1 và 0 trong một số cột

Xử lý các giá trị bị thiếu

```
## Kiểm tra phần trăm dữ liệu bị thiếu trong mỗi cột
(data.isnull().sum()/len(data))*100
```

```
Date             0.000000
Location         0.000000
MinTemp          0.921479
MaxTemp          0.782481
Rainfall         2.023530
Evaporation     38.962731
Sunshine        43.334326
WindGustDir      6.424910
WindGustSpeed    6.368443
WindDir9am       6.556462
WindDir3pm       2.625439
WindSpeed9am     1.096467
WindSpeed3pm     1.900046
Humidity9am      1.646872
Humidity3pm      2.796704
Pressure9am      9.348201
Pressure3pm      9.325242
Cloud9am        34.679871
Cloud3pm        36.833091
Temp9am          1.096467
Temp3pm          2.239473
RainToday        2.023530
RainTomorrow     2.027253
dtype: float64
```

Hình 29: Kiểm tra phần trăm dữ liệu bị thiếu trong mỗi cột

Có nhiều cách xử lý dữ liệu bị thiếu: Sử dụng trung vị hoặc giá trị trung bình, chọn giá trị ngẫu nhiên để đưa vào hoặc giá trị có tần suất xuất hiện nhiều nhất của mỗi thuộc tính

Ở trường hợp dataset này sử dụng điền các giá trị còn thiếu bằng giá trị có tần suất xuất hiện nhiều nhất của mỗi thuộc tính

```
# Điền các giá trị còn thiếu bằng giá trị có tần suất xuất hiện nhiều nhất của mỗi thuộc tính
data['MinTemp']=data['MinTemp'].fillna(data['MinTemp'].mean())
data['MaxTemp']=data['MinTemp'].fillna(data['MaxTemp'].mean())
data['Rainfall']=data['Rainfall'].fillna(data['Rainfall'].mean())
data['Evaporation']=data['Evaporation'].fillna(data['Evaporation'].mean())
data['Sunshine']=data['Sunshine'].fillna(data['Sunshine'].mean())
data['WindGustSpeed']=data['WindGustSpeed'].fillna(data['WindGustSpeed'].mean())
data['WindSpeed9am']=data['WindSpeed9am'].fillna(data['WindSpeed9am'].mean())
data['WindSpeed3pm']=data['WindSpeed3pm'].fillna(data['WindSpeed3pm'].mean())
data['Humidity9am']=data['Humidity9am'].fillna(data['Humidity9am'].mean())
data['Humidity3pm']=data['Humidity3pm'].fillna(data['Humidity3pm'].mean())
data['Pressure9am']=data['Pressure9am'].fillna(data['Pressure9am'].mean())
data['Pressure3pm']=data['Pressure3pm'].fillna(data['Pressure3pm'].mean())
data['Cloud9am']=data['Cloud9am'].fillna(data['Cloud9am'].mean())
data['Cloud3pm']=data['Cloud3pm'].fillna(data['Cloud3pm'].mean())
data['Temp9am']=data['Temp9am'].fillna(data['Temp9am'].mean())
data['Temp3pm']=data['Temp3pm'].fillna(data['Temp3pm'].mean())
```

Hình 30: Điền các giá trị còn thiếu bằng giá trị có tần suất xuất hiện nhiều nhất

```
# Điền các giá trị còn thiếu bằng giá trị có tần suất xuất hiện nhiều nhất của mỗi thuộc tính
data['RainToday']=data['RainToday'].fillna(data['RainToday'].mode()[0])
data['RainTomorrow']=data['RainTomorrow'].fillna(data['RainTomorrow'].mode()[0])
```

Hình 31: Điền giá trị thiếu bằng giá trị có tần suất xuất hiện nhiều nhất

```
# Điền các giá trị còn thiếu bằng giá trị có tần suất xuất hiện nhiều nhất của mỗi thuộc tính
data['WindDir9am'] = data['WindDir9am'].fillna(data['WindDir9am'].mode()[0])
data['WindGustDir'] = data['WindGustDir'].fillna(data['WindGustDir'].mode()[0])
data['WindDir3pm'] = data['WindDir3pm'].fillna(data['WindDir3pm'].mode()[0])
```

Hình 32: Điền giá trị còn thiếu bằng giá trị có tần suất xuất hiện nhiều nhất

Kiểm tra phần trăm dữ liệu bị thiếu trong mỗi cột

```
## Kiểm tra phần trăm dữ liệu bị thiếu trong mỗi cột

(data.isnull().sum()/len(data))*100
```

Hình 33: Kiểm tra phần trăm dữ liệu bị thiếu trong mỗi cột

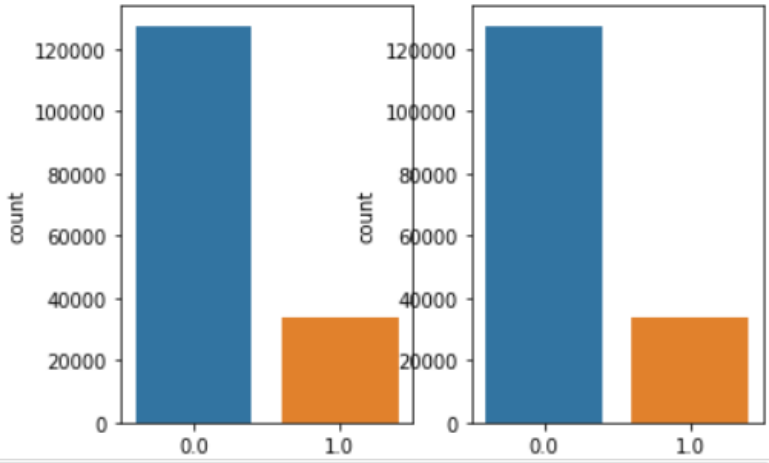
Trực quan hóa dữ liệu

Số lượng raintoday and tomorrow

```
##Số lượng raintoday and raintomorrow
fig, ax =plt.subplots(1,2)
print(data.RainToday.value_counts())
print(data.RainTomorrow.value_counts())

plt.figure(figsize=(20,20))
sns.countplot(data=data,x='RainToday',ax=ax[0])
sns.countplot(data=data,x='RainTomorrow',ax=ax[1])
##Ta sẽ thấy 2 giá trị yes:1 33807 là  giá trị ở rainToday và No:0 là 127347
##Ta sẽ thấy 2 giá trị yes:1 33804 là  giá trị ở RainTomorrow và No:0 là 127350
```

```
0.0    127347
1.0     33807
Name: RainToday, dtype: int64
0.0    127350
1.0     33804
Name: RainTomorrow, dtype: int64
<matplotlib.axes._subplots.AxesSubplot at 0x7f7307f67990>
```



Hình 34: Số lượng raintoday and tomorrow

Xóa cột date để dễ quan xác dữ liệu

Encoding các giá trị categorical

```
#Xóa cột date để dễ quan xác dữ liệu
data=data.iloc[:,1:]
data
```

Hình 35: Xóa cột date để dễ quan xác dữ liệu

```
##Encoding các giá trị categorical
dataset = preprocessing.LabelEncoder()
data['Location'] = dataset.fit_transform(data['Location'])
data['WindDir9am'] = dataset.fit_transform(data['WindDir9am'])
data['WindDir3pm'] = dataset.fit_transform(data['WindDir3pm'])
data['WindGustDir'] = dataset.fit_transform(data['WindGustDir'])
```

Hình 36: Encoding các giá trị categorical

```
data.head(5)
```

	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	WindSpeed9am	WindSpeed3pm	Humidity9am	Humidity3pm	Pressure9am	Pre
0	2	13.4	13.4	0.6	16.077647	7.603032	13	44.0	101	14	20.0	24.0	71.0	22.0	1007.7	
1	2	7.4	7.4	0.0	16.077647	7.603032	14	44.0	94	15	4.0	22.0	44.0	25.0	1010.6	
2	2	12.9	12.9	0.0	16.077647	7.603032	15	46.0	101	15	19.0	26.0	38.0	30.0	1007.6	
3	2	9.2	9.2	0.0	16.077647	7.603032	4	24.0	97	0	11.0	9.0	45.0	16.0	1017.6	
4	2	17.5	17.5	1.0	16.077647	7.603032	13	41.0	89	7	7.0	20.0	82.0	33.0	1010.8	

Hình 37: In ra data sau khi Encoding các giá trị categorical

```
print('Tổng giá trị dữ liệu và thuộc tính trước khi loại bỏ', data.shape )
data=data[(np.abs(stats.zscore(data)) < 3).all(axis=1)]
print('Tổng giá trị dữ liệu và thuộc tính sau khi loại bỏ', data.shape )
```

```
Tổng giá trị dữ liệu và thuộc tính trước khi loại bỏ (161154, 22)
Tổng giá trị dữ liệu và thuộc tính sau khi loại bỏ (145667, 22)
```

Hình 38: Tổng giá trị dữ liệu và thuộc tính trước và sau khi loại bỏ

```
##Bỏ các cột có tương quan cao
data=data.drop(['Temp3pm','Temp9am','Humidity9am'],axis=1)
data.columns
```

```
Index(['Location', 'MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation', 'Sunshine',
       'WindGustDir', 'WindGustSpeed', 'WindDir9am', 'WindDir3pm',
       'WindSpeed9am', 'WindSpeed3pm', 'Humidity3pm', 'Pressure9am',
       'Pressure3pm', 'Cloud9am', 'Cloud3pm', 'RainToday', 'RainTomorrow'],
      dtype='object')
```

Hình 39: Bỏ các cột có tương quan cao

Train và test dữ liệu

```
x_train, x_test, y_train, y_test = train_test_split(data.iloc[:,:-1], data.iloc[:,-1], test_size=0.2, random_state=0)
```

```
x_train.shape## số lượng tập train 80%
```

```
(116533, 21)
```

```
x_test.shape## số lượng tập test 20%
```

```
(29134, 21)
```

Hình 40: Số lượng tập train và test

Sử dụng thuật toán Logistic Regression

```
model = LogisticRegression(max_iter=500)
model.fit(x_train, y_train)
predicted=model.predict(x_test)
conf = confusion_matrix(y_test, predicted)
print ("The accuracy of Logistic Regression is : ", accuracy_score(y_test, predicted)*100, "%")
```

```
The accuracy of Logistic Regression is :  84.00494267865723 %
```

Hình 41: Sử dụng thuật toán Logistic Regression

Sử dụng thuật toán XGBoost

```
xgbc = XGBClassifier(objective='binary:logistic')
xgbc.fit(x_train,y_train)
predicted = xgbc.predict(x_test)
print ("The accuracy of XGBoost is : ", accuracy_score(y_test, predicted)*100, "%")
```

```
The accuracy of XGBoost is :  84.75664172444567 %
```

Hình 42: Sử dụng thuật toán XGBoost

Sử dụng thuật toán Gaussian Naive Bayes

```
model = GaussianNB()
model.fit(x_train, y_train)

predicted = model.predict(x_test)
print("The accuracy of Gaussian Naive Bayes model is : ", accuracy_score(y_test, predicted)*100, "%")
```

```
The accuracy of Gaussian Naive Bayes model is :  80.16750188782865 %
```

Hình 43: Sử dụng thuật toán Gaussian Naive Bayes

Sử dụng thuật toán Bernoulli Naive Bayes

```
model = BernoulliNB()
model.fit(x_train, y_train)

predicted = model.predict(x_test)

print("The accuracy of Bernoulli Naive Bayes model is : ", accuracy_score(y_test, predicted)*100, "%")
```

```
The accuracy of Bernoulli Naive Bayes model is :  76.67330267041945 %
```

Hình 44: Sử dụng thuật toán Bernoulli Naive Bayes

Sử dụng thuật toán Random Forest

```
model = RandomForestRegressor(n_estimators = 100, random_state = 0)
model.fit(x_train, y_train)
predicted = model.predict(x_test)
print("The accuracy of Random Forest is : ", accuracy_score(y_test, predicted.round())*100, "%")
```

```
The accuracy of Random Forest is :  85.48431386009473 %
```

Hình 45: Sử dụng thuật toán Random Forest

4.2 Kết quả

Với đề tài dự báo thời tiết ở nước Úc, quá trình phân tích train và test tập dữ liệu. Với thuật toán Logistic Regression cho ra độ chính xác 84 phần trăm, thuật toán XGBoost 84.75 phần trăm, thuật toán Gaussian Naive Bayes 80.16 phần trăm, thuật toán Bernoulli Naive Bayes 76.67 phần trăm và thuật toán Random Forest Model cho độ chính xác 85.48 phần trăm.

Vậy với chủ đề dự báo thời tiết ở nước Úc thuật toán Random Forest Model cho ra dự đoán chính xác cao nhất. Ngược lại thuật toán Bernoulli Naive Bayes cho ra dự đoán thấp nhất.

Thuật toán Random Forest Model cho ra kết quả chính xác nhất tuy nhiên tốn nhiều thời gian hơn so với các thuật toán còn lại. Thuật toán XGBoost cho ra kết quả dự đoán thời gian ngắn nhất mà độ chính xác so với Thuật toán Random Forest Model cũng không chênh lệch nhiều. Thuật toán Logistic Regression cho ra độ chính xác cao đứng thứ 3 tuy nhiên thời gian dự đoán nhanh mà độ chính xác so với Thuật toán Random Forest Model cũng không chênh lệch nhiều. Thuật toán Gaussian Naive Bayes cùng với thuật toán Bernoulli Naive Bayes cho ra độ chính xác đứng thứ 4 và thứ 5 thời gian dự đoán ngắn tuy nhiên kết quả dự đoán chênh lệch cao.

CHƯƠNG 5 - KẾT LUẬN

Trong bày báo cáo này đã trình bày được cơ sở lý thuyết của thuật toán Logistic Regression , thuật toán Random Forest Model, thuật toán XGBoost , thuật toán Naive Bayes.

Báo cáo nêu được cụ thể từng thuật toán ứng dụng nó vào thực tiễn như nào.

Các ưu nhược điểm của từng thuật toán khi áp dụng nó cho từng bài toán khác nhau.

Với mục đích phân tích và đưa ra dự đoán dự báo thời tiết ở nước Úc thông qua biến RainTomorrow đã đạt được đúng mục đích. So sánh được từng thuật toán nào phù hợp với đề tài.

Trong tương lai sẽ nghiên cứu thực hiện thêm nhiều thuật toán với áp dụng những thuật toán đã tìm hiểu vào nhiều đề tài khác đa dạng hơn.

TÀI LIỆU THAM KHẢO

Tiếng Việt

Tài liệu

[1] Pham Van Chung , (2020, 18 tháng 9) *Logistic Regression - Bài toán cơ bản trong Machine Learning*, Truy xuất từ https://viblo.asia/p/logistic-regression-bai-toan-co-ban-trong-machine-learning

[2] VTI TechBlog! , (2020, 18 tháng 10) *XGBoost – Bài 2: Toàn cảnh về Ensemble Learning – Phần 2*, Truy xuất từ https://vtitech.vn/xgboost-bai-2-toan-canh-ve-ensemble-learning-phan-2

[3] Nguyen Thi Hop , (2019, 14 tháng 9) *Thuật toán phân lớp Naive Bayes*, Truy xuất từ https://viblo.asia/p/thuat-toan-phan-lop-naive-bayes-924lJWPm5PM

[4] Trí tuệ nhân tạo , (2019, 27 tháng 6) *Phần 1: Phân loại Naive Bayes (Lý thuyết)*, Truy xuất từ https://trituenhantao.io/kien-thuc/phan-1-phan-loai-naive-bayes-ly-thuyet/

[5] Machine Learning cơ bản , (2017, 08 tháng 8) *Bài 32: Naive Bayes Classifier*, Truy xuất từ https://machinelearningcoban.com/2017/08/08/nbc/

[6] thuynt , (2018, 21 tháng 7) *Tổng quan về thuật toán phân lớp Naive Bayes Classification (NBC)*, Truy xuất từ http://hoctructuyen123.net/tong-quan-ve-thuat-toan-phan-lop-naive-bayes-classification-nbc/

[7] ICHI.PRO , (2020) *Mọi thứ bạn cần biết về Naïve Bayes*, Truy xuất từ https://ichi.pro/vi/moi-thu-ban-can-biet-ve-naive-bayes-169976819469219

[8] Nam Doan , (2018,tháng 12) *Naive Bayes Classification (NBC) là gì?*, Truy xuất từ https://1upnote.me/post/2018/11/ds-ml-naive-bayes/

Tiếng Anh

Tài liệu

[1] The Pennsylvania State University , (2018) *Logistic Regression*, Truy xuất từ https://online.stat.psu.edu/stat462/node/207/?fbclid=IwAR3BLOZchRa0e8O9ULwAE

[2] ASPER BROTHERS , (2021, 25 tháng 8) *Logistic Regression in Python – Theory and Code Example with Explanation*, Truy xuất từ https://asperbrothers.com/blog/logistic-regression-in-python

[3] Opengenus , (2021) *Advantages and Disadvantages of Logistic Regression*, Truy xuất từ https://iq.opengenus.org/advantages-and-disadvantages-of-logistic-regression/

[4] Christoph Molnar , (2021, 11 tháng 11) *Interpretable Machine Learning*, Truy xuất từ https://christophm.github.io/interpretable-ml-book/logistic.html?fbclid=IwAR2yRWQNclheuFV9h3QlTywBQfdskM38FoWxHhQJ6T

[5] FAHAD MEHFOOZ , (2021, 07 tháng 10) *Rain Prediction with 90.65 accuracy*, Truy xuất từ https://www.kaggle.com/fahadmehfoooz/rain-prediction-with-90-65-accuracy

[6] NIKHIL KHANDELWAL , (2021, 11 tháng 11) *Rain in Australia Prediction*, Truy xuất từ https://www.kaggle.com/nikhilkhandelwal0119/rain-in-australia-prediction