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

BÀI TẬP LỚN GIỮA KÌ MÔN NHẬP MÔN HỌC MÁY

Người hướng dẫn: **GV.LÊ ANH CƯỜNG**

Người thực hiện: **NGUYỄN THANH TÚ – 52100349**

TRẦN THỊ VẸN –52100674

Lớp **: 21050301**

Khoá **: 25**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

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

BÀI TẬP LỚN GIỮA KÌ MÔN NHẬP MÔN HỌC MÁY

Người hướng dẫn: **GV.LÊ ANH CƯỜNG**

Người thực hiện: **NGUYỄN THANH TÚ – 52100349**

TRẦN THỊ VẸN –52100674

Lớp **: 21050301**

Khoá **: 25**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

LỜI CẢM ƠN

Tôi xin gửi lời cảm ơn sâu sắc đến Thầy về việc hướng dẫn và giáo dục trong suốt môn học Máy học trong học kỳ vừa qua. Môn học không chỉ giúp tôi nắm vững các khái niệm cơ bản của Máy học mà còn mở ra cho tôi một thế giới mới về ứng dụng của nó trong thực tế.

Thầy đã tạo ra một môi trường học tập tích cực và truyền cảm hứng cho tôi để tiếp cận với những khía cạnh phức tạp và thú vị của Máy học. Những bài giảng thú vị cùng với ví dụ thực tế đã giúp tôi áp dụng kiến thức một cách hiệu quả trong thực tế.

Tôi cũng rất biết ơn về sự tận tâm và tâm huyết của Thầy trong việc hỗ trợ và giải đáp mọi thắc mắc của chúng tôi. Thầy đã không ngừng khích lệ và động viên chúng tôi để chúng tôi có thể vượt qua những thách thức và phát triển tốt hơn trong môn học này.

Môn học không chỉ giúp tôi nắm vững kiến thức mà còn giúp tôi phát triển kỹ năng phân tích và giải quyết vấn đề một cách chủ động. Tôi hy vọng có cơ hội tiếp tục học hỏi từ Thầy và áp dụng kiến thức đã học được vào thực tế công việc sau này.

Một lần nữa, xin chân thành cảm ơn Thầy về sự hỗ trợ và sự dạy dỗ tận tâm của Thầy.

Trân trọng,

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

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi / chúng tôi và được sự hướng dẫn của GV. Lê Anh Cường;. 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 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 22 tháng 10 năm 2023

Tác giả

(ký tên và ghi rõ họ tên)

Nguyễn Thanh Tú

Trần Thị Vẹn

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

_____

_____

_____

_____

_____

_____

_____

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

_____

_____

_____

_____

_____

_____

_____

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

TÓM TẮT

Câu hỏi 1 : Trình bày và so sánh các mô hình học máy sau: kNN, Linear Regression, Naive Bayes classifiers, Decision Tree. Mỗi mô hình cần phải được phân tích về mục tiêu, phương pháp học, mô hình phù hợp, ưu nhược điểm.

MỤC LỤC

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

CÁC KÝ HIỆU

CÁC CHỮ VIẾT TẮT

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

DANH MỤC HÌNH

DANH MỤC BẢNG

CÂU 1 – PHÂN TÍCH, SO SÁNH CÁC MÔ HÌNH KNN, LINEAR REGRESSION, NAIVE BAYES CLASSIFIERS, DECISION TREE

1.1 Mục tiêu của việc tạo ra mô hình?

Mô hình	Mục tiêu của việc tạo ra mô hình
k-Nearest Neighbors (kNN)	Mô hình máy học dựa trên k-Nearest Neighbors (KNN) là để thực hiện dự đoán hoặc phân loại dựa trên một phương pháp học có giám sát. KNN là một thuật toán đơn giản nhưng hiệu quả được sử dụng cho cả việc dự đoán và phân loại. Các mục tiêu chính của việc tạo một mô hình KNN là như sau: Phân loại: Nếu bạn đang làm việc trên một vấn đề phân loại, mục tiêu là để phân loại hoặc gán nhãn cho các điểm dữ liệu vào các lớp hoặc danh mục đã được xác định trước. KNN có thể phân loại các điểm dữ liệu dựa trên lớp đa số trong số k láng giềng gần nhất. Hồi quy: Nếu bạn đang xử lý một vấn đề hồi quy, mục tiêu là dự đoán một giá trị số liên tục. KNN có thể dự đoán bằng cách trung bình hoặc trọng số giá trị mục tiêu của k láng giềng gần nhất. Ví dụ: Phân loại văn bản: kNN có thể được sử dụng để phân loại tài liệu văn bản dựa trên nội dung của chúng, chẳng hạn như phân loại email là spam hoặc không phải spam. Phân loại ảnh: kNN có thể được sử dụng để phân loại ảnh dựa trên các đặc trưng hình ảnh như màu sắc, hình dáng và kích thước. Lọc cộng đồng: kNN có thể được áp dụng để tạo các hệ thống lọc cộng đồng, ví dụ như gợi ý sản phẩm hoặc nội dung dựa trên hành vi của người dùng khác.
Linear Regression	Mô hình máy học dựa trên Hồi quy tuyến tính là để xây dựng một mối quan hệ dự đoán giữa một tập hợp các đặc trưng đầu vào và một biến mục tiêu liên tục. Hồi quy tuyến tính là một thuật toán học có giám sát được sử dụng cho các nhiệm vụ hồi quy.

	Nó dự đoán một giá trị số liên tục. Các mô hình hồi quy tuyến tính nhằm tìm mối quan hệ tuyến tính giữa các đặc trưng đầu vào và biến mục tiêu. Mối quan hệ này sau đó có thể được sử dụng để thực hiện dự đoán. Ví dụ: Dự đoán giá nhà: Linear Regression thường được sử dụng để dự đoán giá nhà dựa trên các đặc trưng như diện tích, vị trí và số phòng ngủ. Dự đoán doanh số bán hàng: Linear Regression có thể được sử dụng để dự đoán doanh số bán hàng dựa trên quảng cáo hoặc giá cả sản phẩm. Dự đoán thời gian giao hàng: Linear Regression có thể được sử dụng để dự đoán thời gian giao hàng dựa trên khoảng cách và điều kiện giao thông.
Naive Bayes classifiers	Mô hình máy học bằng cách sử dụng các bộ phân loại Naive Bayes là để thực hiện các nhiệm vụ phân loại với sự hỗ trợ của các phương pháp xác suất và thống kê. Naive Bayes là một thuật toán máy học được sử dụng cho nhiệm vụ phân loại. Nó phân loại hoặc xác định các điểm dữ liệu vào các lớp hoặc danh mục được xác định trước. Các mô hình Naive Bayes được sử dụng để dự đoán nhãn lớp của một điểm dữ liệu dựa trên giá trị các đặc trưng của nó. Ví dụ: Phân loại văn bản: Bộ phân loại Naive Bayes thường được sử dụng cho phân loại văn bản, chẳng hạn như phát hiện thư rác (spam) hoặc phân loại tài liệu theo chủ đề. Diagnostico y prediccion en salud: Các bài toán dự đoán và phân loại trong lĩnh vực y tế dựa trên các dữ liệu thực nghiệm và hình ảnh có thể sử dụng mô hình Naive Bayes. Phân loại dữ liệu hạng mục: Naive Bayes có thể được sử dụng để phân loại dữ liệu hạng mục như nguyên liệu trong công nghiệp thực phẩm.
Decision Trees	Mô hình máy học dựa trên cây quyết định (Decision Trees) là tạo ra một mô hình dự đoán hoặc phân loại dữ liệu dựa trên sự học hỏi từ

	các quyết định và quy tắc logic từ dữ liệu huấn luyện. Mô hình cây quyết định là phân loại hoặc dự đoán dữ liệu dựa trên các đặc trưng hoặc thuộc tính của chúng. Mô hình này có khả năng học các quyết định từ dữ liệu huấn luyện và sau đó áp dụng chúng cho dữ liệu mới để xác định lớp hoặc giá trị dự đoán. Ví dụ: Hệ thống quyết định: Decision Trees thường được sử dụng trong việc xây dựng hệ thống quyết định cho các quy trình tổ chức và quản lý. Phân loại văn bản: Decision Trees có thể được sử dụng để phân loại văn bản dựa trên các đặc trưng như từ khóa và nội dung. Dự đoán tỷ lệ thất bại sản phẩm: Trong sản xuất, Decision Trees có thể được sử dụng để dự đoán tỷ lệ thất bại sản phẩm dựa trên các yếu tố sản xuất.

Bảng 1: Mục tiêu của việc tạo ra mô hình

1.2 Phương pháp, giải thuật để học mô hình thế nào, tiêu chí học?

1.2.1 k-Nearest Neighbors (kNN)

Định nghĩa:

K-nearest neighbor (KNN) là một trong những thuật toán học có giám sát đơn giản nhất trong Machine Learning. Ý tưởng của KNN là tìm ra output của dữ kiệu dựa trên thông tin của những dữ liệu training gần nó nhất.

Thuật toán k-NN (với bài toán phân lớp) là như sau:

Bước 1. Nạp tập train và tập test.

Bước 2. Chọn giá trị K thích hợp (xác định tham số K= số láng giềng gần nhất)

Bước 3. Với mỗi điểm một trên tập test:

3.1 Tìm khoảng cách từ điểm đó đến các điểm còn lại trên tập train, sắp xếp từ lớn đến bé, lưu vào danh sách

3.2 Lấy k điểm đầu tiên của danh sách, gắn nhãn chiếm ưu thế trong k điểm cho điểm đang xét.

Bước 4. Dựa vào phần lớn lớp của K để xác định lớp cho đối tượng cần phân lớp. Kết thúc thuật toán.

Chọn số k phù hợp

Giá trị k cho thuật toán k-NN quy định thuật toán phải quan sát k số điểm lân cận xung quanh điểm được xét nhằm phân loại. Nếu k=1, ta gắn nhãn điểm được xét trùng với điểm gần nhất trong tập train. Chọn k là một công việc quan trọng, do các giá trị khác nhau của k có thể dẫn tới hiện tượng over hoặc underfit.

Thường, lựa chọn số k là phụ thuộc vào dữ liệu. Thông thường, giá trị k lớn đồng nghĩa với việc ta giảm thiểu được nhiễu khi thực hiện phân lớp, nhưng sẽ khiến ranh giới giữa các lớp trở nên mờ nhoà hơn. Ta có thể chọn được số k tốt thông qua các thủ thuật heuristic: Scikit-learn cung cấp chức năng GridSearchCV hoặc RandomizeSearchCV, cho phép chúng ta dễ dàng kiểm tra nhiều giá trị k khác nhau. Ngoài 2 thủ thuật trên, ta còn có thể sử dụng cross - validation để có thể lọc k.

Độ chính xác của thuật toán k-NN có thể bị giảm đột ngột nếu có các đặc trưng nhiễu hoặc đặc trưng không liên quan, hoặc scale của các features không đồng đều, không tương xứng với mức độ tương quan với nhãn. Thông thường, khi ta làm việc với bài toán phân lớp nhị phân, ta nên để k lẻ để tránh trường hợp hoà.

Độ đo khoảng cách

Trong k-NN, ta có thể sử dụng nhiều loại khoảng cách khác nhau, nhưng 3 loại khoảng cách thường gặp chính là khoảng cách Manhattan, khoảng cách Euclid và khoảng cách Minkowski.

Khoảng cách Manhattan: là tổng khoảng cách các thành phần toạ độ giữa 2 điểm.

Khoảng cách Euclid: là khoảng cách đường chim bay giữa 2 điểm.

Khoảng cách Minkowski: là một trường hợp tổng quát của cả khoảng cách Euclidean và khoảng cách Chebyshev.

- Độ đo khoảng cách Minkowski

$$d(i,j)=\sqrt[q]{(|x_{i_1}-x_{j_1}|^q+|x_{i_2}-x_{j_2}|^q+...+|x_{i_p}-x_{j_p}|^q)}$$

- Độ đo khoảng cách Manhattan

$$d(i,j)=|x_{i_1}-x_{j_1}|+|x_{i_2}-x_{j_2}|+...+|x_{i_p}-x_{j_p}|$$

- Độ đo khoảng cách Euclidean

$$d(i,j)=\sqrt{(|x_{i_1}-x_{j_1}|^2+|x_{i_2}-x_{j_2}|^2+...+|x_{i_p}-x_{j_p}|^2)}$$

Tiêu chí học của kNN:

Khoảng cách: Tiêu chí quan trọng nhất của KNN là cách tính khoảng cách giữa các điểm dữ liệu. Cách tính khoảng cách ảnh hưởng đến việc xác định láng giềng gần nhất. Phổ biến nhất là khoảng cách Euclidean, nhưng có thể sử dụng các phương pháp khác như khoảng cách Manhattan, khoảng cách Cosine, và nhiều lựa chọn khác tùy thuộc vào bài toán cụ thể.

Giá trị của k: Giá trị k đại diện cho số lượng láng giềng gần nhất mà bạn chọn khi dự đoán hoặc phân loại một điểm dữ liệu mới. Lựa chọn k có thể ảnh hưởng lớn đến hiệu suất của mô hình. Giá trị k quá nhỏ có thể dẫn đến nhiễu, trong khi giá trị k quá lớn có thể làm mất đi tính cục bộ của dự đoán.

Phân loại hoặc hồi quy: KNN có thể được sử dụng cho cả nhiệm vụ phân loại (classification) và dự đoán (regression). Tiêu chí học cụ thể sẽ phụ thuộc vào loại công việc bạn đang thực hiện. Nếu bạn đang phân loại dữ liệu, tiêu chí là xác định lớp của điểm dữ liệu mới dựa trên lớp của láng giềng gần nhất. Nếu bạn đang dự đoán, tiêu chí là tính toán giá trị dự đoán dựa trên giá trị của các láng giềng gần nhất.

Tập dữ liệu huấn luyện: Chất lượng của tập dữ liệu huấn luyện rất quan trọng. Dữ liệu phải được làm sạch và chuẩn bị cẩn thận để đảm bảo rằng các láng giềng gần nhất thực sự liên quan đến dự đoán hoặc phân loại. Nếu dữ liệu không đại diện cho tình huống thực tế hoặc có nhiễu, mô hình KNN có thể không hoạt động tốt.

Sự ánh xạ đặc trưng: Sự ánh xạ và lựa chọn các đặc trưng quan trọng có thể ảnh hưởng đến hiệu suất của mô hình. Đôi khi, việc giảm số lượng đặc trưng có thể cải thiện hiệu suất và giảm độ phức tạp của mô hình.

Trọng số láng giềng: Một phần mở rộng của KNN là sử dụng trọng số cho các láng giềng gần nhất. Cụ thể, các láng giềng có thể được đánh trọng số khác nhau dựa trên khoảng cách hoặc sự quan trọng của chúng trong việc dự đoán. Điều này có thể điều chỉnh hiệu suất của mô hình.

1.2.2 Linear Regression

Định nghĩa:

Hồi quy tuyến tính (Linear Regression) là một thuật toán căn bản nhất đối với bất kì ai bắt đầu học về AI. Trong thực tế, bài toán hồi quy tuyến tính được ứng dụng rất nhiều vì tính dễ dàng mô tả và dễ dàng triển khai. Hồi quy nói chung là lớp bài toán thuộc học có giám sát (Supervised Learning). Dựa trên dữ liệu có sẵn (tức giá trị mục tiêu đã biết) và sự phụ thuộc của giá trị đầu vào để dự đoán một giá trị mới.

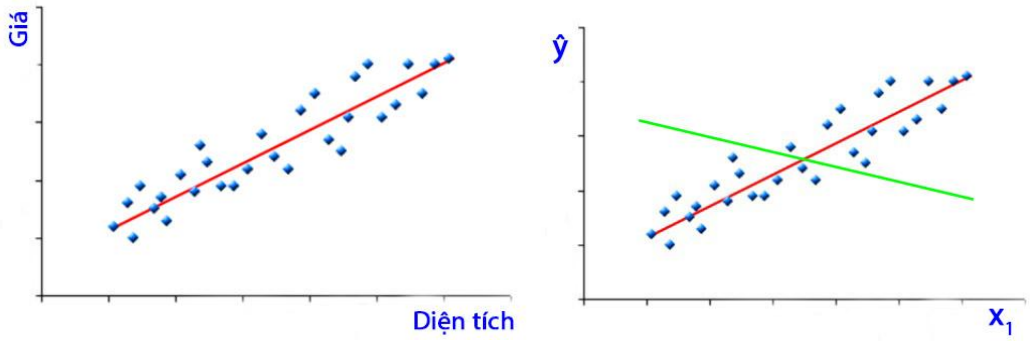
Trong Linear Regression chúng ta sẽ gặp hai loại bài toán đó là Hồi quy đơn biến và Hồi quy đa biến.

Thuật toán của Linear Regression:

Trường hợp đơn giản của thuật toán hồi quy tuyên tính Linear Regression có dạng như sau:

Nếu mỗi input chỉ chứa một biến thì thuật toán được gọi là Linear Regression một biến.

Thuật toán ta cần sử dụng chính là Linear Regression một biến. Khi đó, ta có thể tìm ra được đường thẳng trên hệ trục tọa độ hai chiều biểu diễn sự phụ thuộc giữa giá nhà và diện tích. Ở đồ thị dưới đây, các điểm màu xanh ứng với các input cho trước, các input này chỉ chứa một biến duy nhất là diện tích; đường thẳng cần tìm là đường màu đỏ.



Phương trình đường thẳng trên có thể viết dưới dạng tổng quát như sau

$\widehat{P}$ (s)=w_0+w_1s

với $\widehat{P}$ (s) là giá căn nhà diện tích s.

Lưu ý: Dấu mũ trong $\widehat{P}$ (s) có nghĩa là giá trị này là output được dự đoán từ input nhờ phương trình tìm ra bởi thuật toán. Đối với output thuộc các cặp (input, output) cho trước, ta không dùng dấu mũ. Đây là quy ước chung và được áp dụng mặc định từ giờ trở đi. Do phương trình trên là một hàm tuyến tính và bài toán thuộc loại Regression nên thuật toán được gọi là Linear Regression. Điều này cũng đúng đối với Linear Regression nhiều biến.

Với mỗi cặp giá trị (w_0,w_1) ta được một phương trình đường thẳng khác nhau biểu diễn sự phụ thuộc giữa $\hat{y}$ và x, mỗi phương trình như vậy được gọi là một phương trình giả thuyết (hypothesis function).

Do phương trình giả thuyết dùng để dự đoán giá trị thực tế đối với một input mới nên ta có

y ≈ $\hat{y}$ = x^Tw

Các phương trình giả thuyết khác nhau sẽ có độ chính xác khác nhau. Trong hình dưới đây, rõ ràng đường màu đỏ có độ chính xác cao hơn đường màu xanh khi biểu diễn mối liên hệ giữa $\hat{y}$ và x. Lưu ý là input x chỉ chứa một biến x_1.

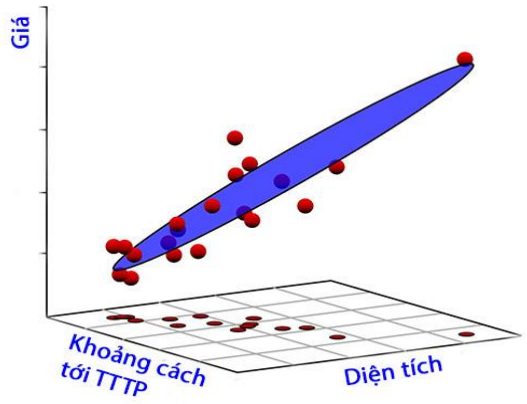
Nhiệm vụ của thuật toán Linear Regression một biến là tìm đường thẳng liên hệ giữa $\hat{y}$ và x hay nói cách khác là tìm ra w - một cách chính xác nhất. Nhưng trước hết, ta cần định nghĩa cụ thể độ chính xác của phương trình giả thuyết là gì đã.

Linear Regression nhiều biến

Nếu mỗi input chứa nhiều hơn một biến thì thuật toán được gọi là Linear Regression nhiều biến.

Trong bài toán tìm giá nhà bằng Linear Regression một biến, ta đã lý tưởng hóa bài toán khi cho rằng giá nhà chỉ phụ thuộc vào diện tích. Trong thực tế, giá nhà còn phụ thuộc vào nhiều yếu tố khác như số phòng, vị trí, thời điểm.

Giả sử giá nhà phụ thuộc vào hai yếu tố là diện tích và khoảng cách tới trung tâm thành phố, khi đó thuật toán Linear Regression với hai biến có nhiệm vụ tìm ra mặt phẳng biểu diễn sự phụ thuộc giữa giá nhà với hai yếu tố trên một cách chính xác nhất.



Phương trình mặt phẳng trên có thể viết dưới dạng tổng quát như sau

$\widehat{P}$ (s,d)=w_0+w_1s+w_2d

với $\widehat{P}$ (s,d) là giá căn nhà diện tích s và cách trung tâm thành phố khoảng cách d.

Trong trường hợp tổng quát, input bao gồm n biến x_1,x_2,…,x_n và output $\hat{y}$ phụ thuộc vào n biến đó theo phương trình tuyến tính

$\hat{y}$ = w_0+w_1x_1+…+w_n−1x_n−1+w_nx_n

thì thuật toán tìm ra phương trình ở trên gọi là Linear Regression n biến.

Để đơn giản hóa phương trình ở trên, ta có thể đặt w = $\begin{bmatrix} w_0 \\ w_1 \\ \cdots \\ w_n \end{bmatrix}$ là vector các hệ số của phương trình, x = $\begin{bmatrix} 1 \\ x_1 \\ \cdots \\ x_n \end{bmatrix}$ là vector các biến của input (phần tử 1 trong vector đóng vai trò như x_0 chỉ nhằm mục đích thuận tiện cho tính toán) thì có thể viết lại phương trình trên như sau:

$\hat{y} = x^T w$

Độ chính xác của phương trình giả thuyết:

Nói chung, ta không thể tìm được một đường thẳng đi qua tất cả các điểm input cho trước được (trừ khi chúng thẳng hàng). Do vậy với mỗi phương trình giả thuyết luôn có sự mất mát nhất định. Độ lớn của sự mất mát phụ thuộc các tham số (w_0,w_1) và được tính bằng phương trình hàm mất mát (cost function) sau:

$$J(w_0, w_1) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)})^2$$

$$= \frac{1}{2m} \sum_{i=1}^{m} (w_0 + w_1 x_1^{(i)} - y^{(i)})^2$$

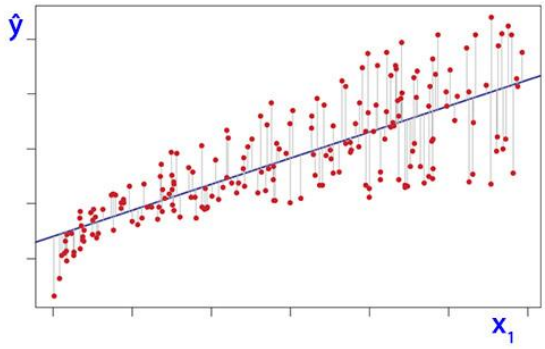
$$= \frac{1}{2m} \sum_{i=1}^{m} (x^{(i)T} w - y^{(i)})^2$$

trong đó m là số input ban đầu được dùng để đào tạo thuật toán; ký hiệu $x_1^{(2)}$ có nghĩa là giá trị biến x1 thuộc vào input thứ 2 trong bộ input ban đầu để đào tạo thuật toán. Các quy ước về ký hiệu này sẽ được sử dụng mặc định từ giờ trở đi.

Một cách hình học, hàm mất mát bằng một nửa trung bình cộng bình phương các khoảng cách sai lệch trong hình dưới đây:

Vector các output thì hàm mất mát được viết lại thành

$$J(w) = \frac{1}{2m} \| Xw - y \|^2$$



Tiêu chí học của Linear Regression:

Hàm Mất Mát (Loss Function): Đây là một hàm đo lường sự sai khác giữa dự đoán của mô hình và giá trị thực tế trong dữ liệu huấn luyện. Trong Linear Regression, hàm mất mát thường là Mean Squared Error (MSE), tức là tổng bình phương sai khác giữa dự đoán và giá trị thực tế được chia cho số mẫu.

Phương pháp tối ưu hóa (Optimization Method): Để tìm trọng số tối ưu, cần áp dụng một phương pháp tối ưu hóa như Gradient Descent. Phương pháp này cố gắng điều chỉnh trọng số để giảm thiểu hàm mất mát. Trong quá trình học, mô hình điều chỉnh trọng số theo gradient âm của hàm mất mát.

Tiêu chuẩn dừng (Stopping Criteria): Quá trình học sẽ dừng khi một tiêu chuẩn dừng được đáp ứng. Điều này có thể là một số lượng vòng lặp tối đa, một ngưỡng cho hàm mất mát hoặc sự hội tụ của trọng số.

Regularization (Chính quy hóa): Trong một số trường hợp, Linear Regression có thể sử dụng chính quy hóa để kiểm soát overfitting. Có hai loại phổ biến là L1 regularization (Lasso) và L2 regularization (Ridge).

1.2.3 Naive Bayes classifiers

Định nghĩa:

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 Classification 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. Nó thuộc vào nhóm Supervised Machine Learning Algorithms.

Thuật toán Naive Bayes Classification:

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 lí 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. Đây được gọi là xác suất biên duyên hay xác suất tiên nghiệm, nó là "tiên nghiệm" 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 đang muốn biết.

* 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. Chú ý không nhầm lẫn giữa khả năng xảy ra B khi biết A và xác suất xảy ra A khi biết B.

Công thức chỉ ra xác suất của A xảy ra nếu B cũng xảy ra, ta viết là P(A|B). Và nếu ta biết xác suất của B xảy ra khi biết A, ta viết là P(B|A) cũng như xác suất độc lập của A và B.

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:

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

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

Công thức 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:

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

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

P(A|B) được gọi là "xác suất của A khi biết B" và thường được gọi là xác suất hậu nghiệm (posterior probability) của A khi có thông tin từ B.

P(B|A) cũng được gọi là xác suất hậu nghiệm (posterior probability) của B với điều kiện A.

P(A) và P(B) được gọi là xác suất tiền nghiệm (prior probability) của A và B, tương ứng. Đây là xác suất trước khi có bất kỳ thông tin mới nào từ sự kiện khác (A hoặc B).

Có ba loại Mô hình Naive Bayes được đưa ra dưới đây:

Gaussian: Mô hình Gaussian giả định rằng các đặc tính tuân theo phân phối chuẩn. Điều này có nghĩa là nếu các yếu tố dự đoán lấy các giá trị liên tục thay vì rời rạc thì mô hình giả định rằng các giá trị này được lấy mẫu từ phân bố Gaussian.

Multinomial: Trình phân loại Naïve Bayes đa thức được sử dụng khi dữ liệu được phân phối đa thức. Nó chủ yếu được sử dụng cho các bài toán phân loại tài liệu, có nghĩa là một tài liệu cụ thể thuộc về danh mục nào, chẳng hạn như Thể thao, Chính trị, giáo dục, v.v. Trình phân loại sử dụng tần suất các từ để dự đoán.

Bernoulli: Trình phân loại Bernoulli hoạt động tương tự như Trình phân loại đa thức, nhưng các biến dự đoán là các biến Booleans độc lập. Chẳng hạn như một từ cụ thể có hiện diện hay không trong tài liệu. Mô hình này cũng nổi tiếng với nhiệm vụ phân loại tài liệu.

Các bước thực hiện:

Bước 1: Ước tính xác suất: Naive Bayes dựa trên ước tính xác suất để dự đoán hoặc phân loại dữ liệu. Nó sử dụng lý thuyết Bayes để tính xác suất của một sự kiện dựa trên xác suất của các sự kiện trước đó.

Bước 2: Giả định đơn giản: "Naive" trong tên của mô hình Naive Bayes ám chỉ một giả định đơn giản, đó là các biến đầu vào độc lập có điều kiện, nghĩa là mối quan hệ giữa các biến đầu vào được giả định là độc lập. Dựa vào giả định này, mô hình đơn giản hóa quá trình tính toán xác suất.

Bước 3: Xác định xác suất tiền nghiệm: Naive Bayes xác định xác suất tiền nghiệm (prior probability) của các lớp hoặc sự kiện từ dữ liệu huấn luyện. Xác suất tiền nghiệm là xác suất ban đầu mà chúng ta biết trước về các lớp hoặc sự kiện.

Bước 4: Xác định xác suất hậu nghiệm: Naive Bayes sử dụng dữ liệu huấn luyện để xác định xác suất hậu nghiệm (posterior probability) của các lớp hoặc sự kiện, dựa trên xác suất tiền nghiệm và thông tin từ dữ liệu huấn luyện.

Ví dụ, khi bạn áp dụng mô hình Naive Bayes để phân loại email là "spam" hoặc "không spam"

Tính posterior probability P(spam|email) để xác định xác suất một email cụ thể là "spam" sau khi đã xem xét xác suất tiền nghiệm (P(spam)) và xác suất ước tính (likelihood P(email|spam)). Posterior probability P(spam|email) sẽ cho bạn biết xác suất một email cụ thể thuộc vào lớp "spam" sau khi đã xem xét thông tin từ email đó.

Tiêu chí học của Naive Bayes Classification:

Xác suất tối đa: Tiêu chí học của Naïve Bayes là tìm ra các xác suất hậu nghiệm tối đa dựa trên xác suất tiền nghiệm và thông tin từ dữ liệu huấn luyện. Mô hình cố gắng tìm ra lớp hoặc sự kiện có xác suất cao nhất dựa trên thông tin có sẵn.

Dữ liệu huấn luyện: Mô hình Naïve Bayes cần một tập dữ liệu huấn luyện chứa các ví dụ về các biến đầu vào và các lớp hoặc sự kiện tương ứng để học xác suất.

Ứng dụng trong phân loại và dự đoán: Naïve Bayes thường được sử dụng cho các tác vụ phân loại (classification) và dự đoán, trong đó mô hình cố gắng phân loại hoặc dự đoán lớp hoặc sự kiện dựa trên thông tin từ biến đầu vào.

Kiểm tra và đánh giá: Để đảm bảo tính chính xác của mô hình, tiêu chí học của Naïve Bayes bao gồm việc kiểm tra và đánh giá mô hình trên tập dữ liệu kiểm tra để đảm bảo rằng nó hoạt động tốt trên dữ liệu mới.

1.2.4 Decision Trees

Định nghĩa:

Cây quyết định (Decision Tree) là một kỹ thuật học máy giám sát được sử dụng cho cả bài toán phân loại và hồi quy, nhưng thường được ưa chuộng trong việc giải quyết bài toán phân loại. Nó là một loại bộ phân loại được biểu diễn dưới dạng cây, trong đó các nút nội tại (internal nodes) biểu thị các đặc điểm (features) của bộ dữ liệu, các nhánh biểu thị các quy tắc quyết định và mỗi nút lá (leaf node) biểu thị kết quả cuối cùng.

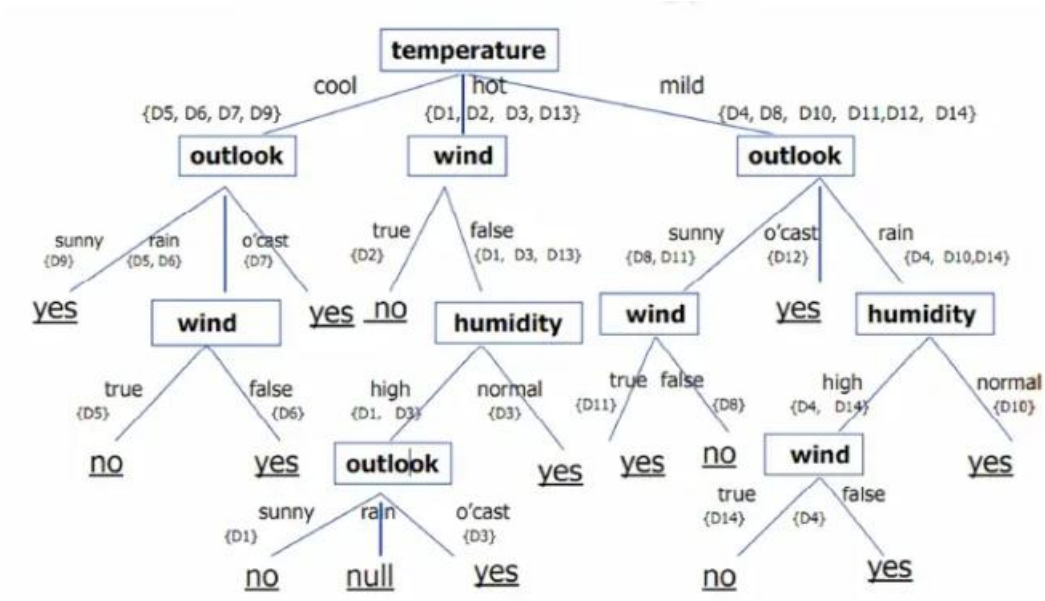
Decision Trees gồm 3 phần chính: 1 node gốc (root node), những node lá (leaf nodes) và các nhánh của nó (branches). Node gốc là điểm bắt đầu của cây quyết định và cả hai node gốc và node chứa câu hỏi hoặc tiêu chí để được trả lời. Nhánh biểu diễn các kết quả của kiểm tra trên nút.

Ví dụ câu hỏi ở node đầu tiên yêu cầu câu trả lời là “yes” hoặc là “no” thì sẽ có 1 node con chịu trách nhiệm cho phản hồi là “yes”, 1 node là “no”.

Chiến lược cơ bản để xây dựng cây quyết định:

- Bắt đầu từ nút đơn biểu diễn tất cả các mẫu
- Nếu các mẫu thuộc về cùng một lớp, nút trở thành nút lá và được gán nhãn bảng lớp đó

- Ngược lại, dùng độ đo thuộc tỉnh để chọn thuộc tính sẽ phân tách tốt nhất các mẫu vào các lớp
- Một nhánh được tạo cho từng giá trị của thuộc tính được chọn và các mẫu được phân hoạch theo
- Dùng để quy cùng một quá trình để tạo cây quyết định.
- Tiến trình kết thúc chi khi bất kỳ điều kiện nào sau đây là đúng
    - Tất cả các mẫu cho một nút cho trước đều thuộc về cùng một lớp.
    - Không còn thuộc tỉnh nào mà mẫu có thể dựa vào để phân hoạch xa hơn.
    - Không còn mẫu nào cho nhánh test_attribute = a_i
- Tuy nhiên, nếu không chọn được thuộc tỉnh phân lớp hợp lý tại mỗi nút, ta sẽ tạo ca cây rất phức tạp, ví dụ như cây dưới đây.



Thuật toán liên quan đến Decision Trees:

ID3 → (Iterative Dichotomiser 3)

C4.5 → (kế thừa của ID3)

CART → (Cây phân loại và hồi quy)

CHAID → (Phát hiện tương tác tự động chi bình phương Thực hiện phân chia nhiều cấp khi tính toán cây phân loại)

MARS → (splines hồi quy thích ứng đa biến)

Conditional Inference Trees

Thuật toán ID3

ID3 (J. R. Quinlan 1993) sử dụng phương pháp tham lam tìm kiếm từ trên xuống thông qua không gian của các nhánh có thể không có backtracking. ID3 sử dụng Entropy và Information Gain để xây dựng một cây quyết định.

Có rất nhiều hệ số khác nhau mà phương pháp cây quyết định sử dụng để phân chia. Dưới đây, tôi sẽ đưa ra hai hệ số phổ biến là Information Gain và Gain Ratio (ngoài ra còn hệ số Gini).

Entropy trong Cây quyết định (Decision Tree)

Entropy là thuật ngữ thuộc Nhiệt động lực học, là thước đo của sự biến đổi, hỗn loạn hoặc ngẫu nhiên. Năm 1948, Shannon đã mở rộng khái niệm Entropy sang lĩnh vực nghiên cứu, thống kê với công thức như sau:

Với một phân phối xác suất của một biến rời rạc x có thể nhận n giá trị khác nhau x_1, x_2, …, x_n

Giả sử rằng xác suất để x nhận các giá trị này là $p_i=p(x=x_i)$

Ký hiệu phân phối này là $p = (p_1, p_2, …, p_n)$. Entropy của phân phối này được định nghĩa là:

$$H(p)= – \sum^{n}_{n=1} p_i \log(p_i)$$

Giả sử bạn tung một đồng xu, entropy sẽ được tính như sau:
$H = - [0.5 \ln (0.5) + 0.5 \ln (0.5)]$

1.3 Mô hình phù hợp cho loại bài toán và dữ liệu nào, ưu nhược điểm?

Mô hình	Problem and data types	Ưu điểm	Nhược điểm
KNN	**Problem types:** Phân loại và hồi quy. Nhưng chủ yếu dùng phân loại. **Data types:** Dữ liệu dạng số (numerical data): kNN thường được sử dụng với dữ liệu số học, ví dụ: các đặc trưng định lượng như chiều cao, cân nặng, khoảng cách, vv. Dữ liệu dạng hạng mục (categorical data): kNN cũng có thể sử dụng với dữ liệu hạng mục, nhưng bạn cần chuyển đổi chúng thành dạng số (one-hot encoding) trước. Đặc biệt trong trường hợp dữ liệu không tuyến tính và có biên quyết định phức tạp. Nó cũng hiệu quả khi dữ liệu không quá lớn.	1. Độ phức tạp tính toán của quá trình training là bằng 0. 2. Việc dự đoán kết quả của dữ liệu mới rất đơn giản. 3. Không cần giả sử gì về phân phối của các class. 4. Thuật toán đơn giản, dễ dàng triển khai. 5. Độ phức tạp tính toán nhỏ. 6. Xử lý tốt với tập dữ liệu nhiễu	1. Với K nhỏ dễ gặp nhiễu dẫn tới kết quả đưa ra không chính xác. 2. Cần nhiều thời gian để thực hiện do phải tính toán khoảng cách với tất cả các đối tượng trong tập dữ liệu. 3. Cần chuyển đổi kiểu dữ liệu thành các yếu tố định tính. 4. Tốn nhiều bộ nhớ, việc lưu toàn bộ dữ liệu trong bộ nhớ cũng ảnh hưởng tới hiệu năng của KNN.
Linear Regression	**Problem types:** Hồi quy **Data types:** Dữ liệu dạng số (numerical data): Linear Regression phù hợp với các vấn đề hồi quy khi	1. Nhanh chóng và đơn giản cho việc mô hình hóa, đặc biệt hiệu quả khi mối quan hệ không quá phức tạp và có	1. Dễ bị ảnh hưởng bởi giá trị ngoại lệ trong dữ liệu. 2. Dễ bị ảnh hưởng nếu có nhiều đặc trưng tương

	biến đầu vào và biến mục tiêu là dạng số liên tục. Hồi quy tuyến tính phù hợp cho các vấn đề hồi quy khi mối quan hệ giữa các đặc trưng đầu vào và biến mục tiêu là gần như tuyến tính.	ít dữ liệu. 2. Đơn giản và dễ hiểu, hữu ích cho quyết định kinh doanh. 3. Có hệ số dễ hiểu, kèm theo khoảng tin cậy và kiểm tra thống kê, quan trọng cho suy luận. 4. Không cần điều chỉnh siêu tham số, giảm phức tạp trong việc điều chỉnh mô hình. 5. Được biết đến rộng rãi, giúp xây dựng sự tin tưởng từ các bên liên quan. 6. Suy luận nhanh chóng và đơn giản, dễ triển khai trong sản xuất mà không cần thư viện học máy phức tạp.	quan cao. 3. Cần xác định rõ ràng các tương tác giữa các đặc trưng. 4. Giả định mối quan hệ tuyến tính giữa đặc trưng và kết quả. 5. Không thể xử lý dữ liệu bị thiếu một cách tự nhiên. 6. Hiệu suất dự đoán thường không cao nhất trên dữ liệu bảng.
Naive Bayes classifiers	**Problem types:** Phân loại **Data types:** Dữ liệu hạng mục (categorical data): Naive Bayes thường được sử	1. Giả định độc lập: hoạt động tốt cho nhiều bài toán/miền sữ liệu và ứng dụng. Đơn	1. Mô hình không được huấn luyện bằng phượng pháp tối ưu mạnh và chặt

	dụng cho các vấn đề phân loại văn bản và phát hiện thư rác (spam detection) trong đó các đặc trưng có thể là các từ hoặc hạng mục (ví dụ: "spam" hoặc "non-spam"). Dữ liệu đếm (count data): Naive Bayes cũng hoạt động tốt cho các vấn đề đếm, như đếm số lượng từ trong văn bản. Bộ phân loại Naive Bayes thường được sử dụng cho các vấn đề phân loại văn bản, phát hiện thư rác và các vấn đề phân loại khác liên quan đến dữ liệu phân loại hoặc rời rạc.	giản nhưng đủ tốt để giải quyết nhiều bài toán như phân lớp văn bản, lọc spam,.. 2. Cho phép kết hợp tri thức tiền nghiệm (prior knowledge) và dữ liệu quan sát được (obserwed data).Tốt khi có sự chệnh lệch số lượng giữa các lớp phân loại. 3. Huấn luyện mô hình (ước lượng tham số) dễ và nhanh. 4. Đơn giản và ít phức tạp: So với các bộ phân loại khác, Naive Bayes được xem là một bộ phân loại đơn giản vì tham số dễ dàng ước tính. Do đó, nó thường là một trong những thuật toán đầu tiên	chẽ. 2. Tham số mủa mô hình là các ước lượng xác suất điều kiện đơn lẻ. 3. Không tính đến sự tương tác giữa các ước lượng này. 4. Giả định về tính độc lập: Naive Bayes giả định rằng tất cả các biến dự đoán (hoặc đặc trưng) là độc lập, điều hiếm gặp trong thực tế. Điều này hạn chế khả năng áp dụng của thuật toán trong các tình huống thực tế. 5. Vấn đề 'zero-frequency': Thuật toán này đối mặt với vấn đề "zero-frequency" khi gán xác suất bằng không cho biến hạng mục mà không có trong tập dữ liệu huấn luyện. Điều

		được học trong khoá học về khoa học dữ liệu và học máy. 5. Xử lý dữ liệu có số chiều cao: Trường hợp sử dụng, chẳng hạn như phân loại tài liệu, có thể có số chiều dữ liệu cao, điều này có thể khó khăn cho các bộ phân loại khác. Naive Bayes có thể xử lý tốt dữ liệu có số chiều cao như vậy.	này yêu cầu sử dụng kỹ thuật smoothing (làm mượt) để khắc phục vấn đề này. 6. Ước tính có thể sai lệch: Trong một số trường hợp, Naive Bayes có thể đưa ra các ước tính sai lệch, do đó không nên chấp nhận quá mức các đầu ra xác suất của nó.
Decision Tree	**Problem types:** **Data types:** Dữ liệu dạng số (numerical data): Cây quyết định hoạt động với cả dữ liệu số học và dữ liệu hạng mục. Dữ liệu hạng mục (categorical data): Các biến hạng mục có thể được sử dụng trực tiếp trong cây quyết định sau khi chuyển đổi thành dạng one-hot encoding. Cây quyết định có tính đa dạng và có thể sử dụng cho cả vấn đề phân loại	1. Mô hình dễ hiểu và dễ giải thích, nó rất có giá trị cho các quyết định kinh doanh. 2. Cần ít dữ liệu để huấn luyện. Có thể xử lý tốt với dữ liệu dạng số (rời rạc và liên tục) và dữ liệu hạng mục. Có thể giải quyết tốt các vấn đề phi tuyến tính.	1. Khó trong việc tính toán entropy và information gain. 2. Có thể overfitting (tạo ra những cây quá khớp với dữ liệu huấn luyện hay quá phức tạp). 3. Thường ưu tiên thuộc tính có nhiều giá trị (khắc phục bằng cách sử

	và hồi quy. Chúng hiệu quả trong các tình huống dữ liệu có mối quan hệ phức tạp và phi tuyến tính.	3. Mô hình dạng white box rõ ràng. Xây dựng nhanh. Phân lớp nhanh. 4. Cây quyết định không đòi hỏi việc chuẩn hóa dữ liệu, không đòi hỏi việc tỉ lệ hóa dữ liệu. 5. Các giá trị bị thiếu trong dữ liệu cũng KHÔNG ảnh hưởng đáng kể đến quá trình xây dựng cây quyết định. 6. Mô hình cây quyết định rất dễ hiểu và giải thích cho các nhóm kỹ thuật cũng như các bên liên quan.	dụng Gain Ratio) 4. Sự không ổn định: Một thay đổi nhỏ trong dữ liệu có thể dẫn đến một thay đổi lớn trong cấu trúc của cây quyết định, gây sự không ổn định. Không đảm bảo xây dựng được cây tối ưu. 5. Phức tạp tính toán: Đôi khi, tính toán trong cây quyết định có thể trở nên phức tạp hơn so với các thuật toán khác. 6. Thời gian đào tạo lâu: Quá trình đào tạo mô hình cây quyết định thường mất nhiều thời gian hơn. 7. Việc đào tạo cây quyết định đòi hỏi nhiều thời gian và tài nguyên tính toán, do đó

			tương đối đắt đỏ. 8. Không thích hợp cho hồi quy: Thuật toán cây quyết định thường không phù hợp cho việc áp dụng hồi quy và dự đoán giá trị liên tục.

Bảng 2. Loại bài toán và dữ liệu, ưu nhược điểm các mô hình

TÀI LIỆU THAM KHẢO

Tiếng Việt

Tiếng Anh

PHỤ LỤC