BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH
KHOA CÔNG NGHỆ THÔNG TIN

TIỂU LUẬN KHAI THÁC DỮ LIỆU VÀ ỨNG DỤNG

DỰ BÁO THỜI TIẾT VỚI DỮ LIỆU CHUỖI THỜI GIAN VÀ SO SÁNH GIỮA LSTM VÀ RNN

Giảng viên hướng dẫn : **TS Võ Thị Hồng Thắm**
Sinh viên thực hiện : **Phan A Hảo**
MSSV : **2200005484**
Khóa : **22**
Chuyên ngành : **Trí tuệ nhân tạo**

TP. Hồ Chí Minh, tháng 06, năm 2024

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH
KHOA CÔNG NGHỆ THÔNG TIN

TIỂU LUẬN KHAI THÁC DỮ LIỆU VÀ ỨNG DỤNG

DỰ BÁO THỜI TIẾT VỚI DỮ LIỆU CHUỖI THỜI GIAN VÀ SO SÁNH GIỮA LSTM VÀ RNN

Giảng viên hướng dẫn : **TS Võ Thị Hồng Thắm**
Sinh viên thực hiện : **Phan A Hảo**
MSSV : **2200005484**
Khóa : **22**
Chuyên ngành : **Trí tuệ nhân tạo**

TP. Hồ Chí Minh, tháng 06, năm 2024

BM-ChT-

TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH
TRUNG TÂM KHẢO THÍ

KỲ THI KẾT THÚC HỌC PHẦN
HỌC KỲ 2 NĂM HỌC 2023 - 2024

PHIẾU CHẤM THI TIỂU LUẬN

Môn thi: **Khai thác dữ liệu và ứng dụng** Lớp học phần: **21DTH2C**
Nhóm sinh viên thực hiện :**Phan A Hảo** Tham gia đóng góp: **Toàn bộ**
Ngày thi: **06/06/2024** Phòng thi:**L.904**
Đề tài tiểu luận/báo cáo của sinh viên : Dự báo thời tiết với dữ liệu chuỗi thời gian và so sánh giữa LSTM và RNNs.
Phần đánh giá của giảng viên (căn cứ trên thang rubrics của môn học):

Tiêu chí (theo CĐR HP)	**Đánh giá của GV**	**Điểm tối đa**	**Điểm đạt được**
Cấu trúc của báo cáo	....		
Nội dung	....		
- Các nội dung thành phần	....		
- Lập luận	....		
- Kết luận	.....		
Trình bày	....		
TỔNG ĐIỂM			

Giảng viên chấm thi
(ký, ghi rõ họ tên)

TS. Võ Thị Hồng Thắm

NHẬN XÉT CỦA GIẢNG VIÊN

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

LỜI MỞ ĐẦU

Hiện nay, chúng ta đang sống trong thời đại của dữ liệu, hàng ngày có khoảng Terabytes hoặc Petabytes dữ liệu được thu thập từ các doanh nghiệp, y tế, khoa học và kỹ thuật… Sự bùng nổ về data này là kết quả của việc số hóa bằng các công cụ thu thập và lưu trữ mạnh mẽ. Với một lượng dữ liệu khổng lồ như vây, thay vì đưa ra quyết định dựa trên cá nhân hiện nay đưa quyết định dựa trên phân tích khai phá dữ liệu đang dần thay thế. Nhưng với một lượng dữ liệu khổng lồ được tạo ra mỗi ngày đòi hỏi các công cụ xử lý, khai phá tri thức từ dữ liệu phải liên tục được phát triển và cập nhật. Sự cần thiết này dẫn đến sự ra đời của vô số công cụ và là một bước tiến lớn trong lịch sử phát triển nhân loại. Là một sinh viên chuyên ngành Trí tuệ nhân tạo, tôi rất vinh dự khi đóng góp một phần tri thức của tôi vào quá trình nghiêng cứu, phân tích ở lĩnh vựa được coi là đầy hứa hẹn như phân tích và khai phá dữ liệu.

LỜI CẢM ƠN

Tôi xin bày tỏ lòng biết ơn chân thành và sâu sắc đến Giảng viên TS. Võ Thị Hồng Thắm đã truyền đạt kiến thức, kinh nghiệm, và sự định hướng trong suốt quá trình thực hiện dự án này. Sự hỗ trợ tận tình, những kinh nghiệm và khuyến khích từ phía cô đã giúp tôi vượt qua những thách thức, khó khăn và phát triển không ngừng trong lĩnh vực nghiêng cứu và khai phá dữ liệu.

Tôi cũng muốn gửi lời cảm ơn đặc biệt tới bạn bè, người thân và những người đã cùng chia sẻ những ý tưởng và phản hồi quý báu. Sự đóng góp của bạn đã là nguồn động viên vô cùng quan trọng để tôi tiến xa hơn trên con đường này.

Tôi cũng không thể không đề cập đến những tài liệu, công cụ và nguồn thông tin mà chúng tôi đã sử dụng để nắm vững kiến thức cơ bản và tiến xa hơn trong việc phát triển các mô hình trí tuệ nhân tạo.

Xin chân thành cảm ơn và hy vọng rằng công trình này sẽ tạo động lực cho thế hệ sau, tiếp tục góp phần nhỏ vào sự phát triển của lĩnh vực nghiêng cứu, khai phá dữ liệu và cộng đồng người sử dụng công nghệ.

MỤC LỤC

DANH MỤC HÌNH

KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT

Chữ viết tắt	Ý nghĩa
LSTM	Long Short Term Memory
RNN	Recurrent Neural Network
MAE	Mean Absolute Error
MSE	Mean Squared Error
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
OPTICS	Ordering Points to Identify the Clustering Structure
DENCLUE	DENsity-based CLUstEring
ID3	Iterative Dichotomiser
exp	example

Chương 1. TỔNG QUAN ĐỀ TÀI

1.1 Lý do chọn đề tài

Ngày nay được gọi là thời đại của dữ liệu, một thời đại mà mọi quyết định đều dựa trên dữ liệu. Các ứng mô hình sử dụng data để hỗ trợ ra quyết định đang ngày càng đi sâu vào đời sống con người. Con người cũng dần quen thuộc các công cụ trên và sử dụng chũng một cách rộng rãi.

Với lượng dữ liệu khổng lồ nhận được về thời tiết mỗi ngày và tính chất liên tục của thời tiết. Một mô hình tận dụng lượng dữ liệu khổng lồ trên để phân tích và đưa ra các dự đoàn về nhiệt độ, lượng mưa… là cần thiết. Thay vì đưa ra dự đoán dựa trên quát sát, kinh nghiệm như ngày xưa thì dự đoán bằng mô hình xây dựng dựa trên dữ liệu mang tính khách quan hơn.

Mô hình dự đoán nhiệt độ ra đời dựa trên sự cần thiết và một môi trường ngập tràn dữ liệu. Mô hình sử dụng nhiều thuộc tính của thời tiết từ nhiều nơi, nhiều mốc thời gian, nhận vào các thuộc tính tương tự và đưa ra mức nhiệt độ dự đoán chính xác nhất có thể của ngày hôm nay

Từ những vấn đề đặt ra ở trên, được sự đồng ý và sự hướng dẫn tận tình của TS. Võ Thị Hồng Thắm, tôi đã chọn đề tài: “Dự báo thời tiết với dữ liệu chuỗi thời gian và so sánh giữa LSTM và RNN”. Hy vọng đề tài sẽ được đưa vào ứng dụng trong thực tế.

1.2 Môi trường phát triển và các công cụ hỗ trợ

Toàn bộ mô hình sẽ được tạo bằng ngôn ngữ Python và thư viện Pytorch và một số thư viện liên quan như pyplot để vẽ đồ thị, sklearn để chuẩn hóa, mô hình là 2 loại mạng neural tái phát là RNNs và LSTM.

Bộ dữ liệu thời tiết của Thành Phố Hồ Chí Minh được thu thập trên Internet.

1.3 Cấu trúc đề tài

Để trình bày nghiên cứu một cách logic và dễ hiểu, tôi đề xuất bố cục cho đề tài " Dự báo thời tiết với dữ liệu chuỗi thời gian và so sánh giữa LSTM và RNN " như sau:

Giới thiệu:

Lý do chọn đề tài

Phương pháp và phạm vi nghiên cứu

Môi trường phát triển và các công cụ hỗ trợ.

Cơ sở lý thuyết:

Tổng quan về Data Mining

Tiền xử lý dữ liệu.

Các bài toán ứng dụng như: Luật kết hợp, Phân lớp, Dự đoán, Gom cụm.

Xu hướng của Data Mining.

Phân tích xây dựng hệ thống.

Phân tích cấu trúc hệ thống

Tiền xử lý dữ liệu

Triển khai, kiểm tra và so sánh

Kết luận.

Tích hợp các kết quả đạt được.

Đề xuất hướng phát triển và cải tiến tương lai.

Chương 2. CƠ SỞ LÝ THUYẾT

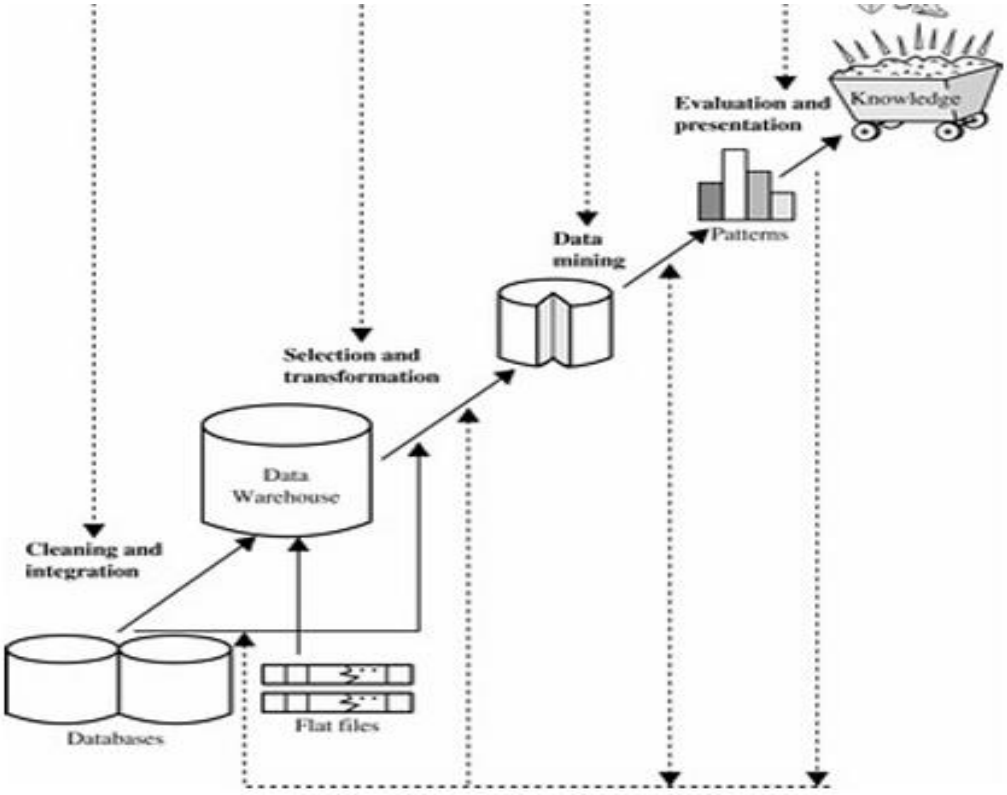
2.1 Tổng quan về data mining và quy trình khai phá tri thức

2.1.1 Khái niệm Data Mining.

Khái niệm Data Mining hay còn gọi là khai thác dữ liệu. Khái niệm này chỉ mô tả ngắn gọn, thực chất Data Mining được gọi là khai thách và sử dụng kiến thức từ dữ liệu. Data Mining là 1 bước trong cả một quá trình từ dữ liệu thô sang kiến thức được trình diện cho người dùng, trong quy trình đó Data Mining tiếp nhận dữ liệu đã được xử lý, thông qua các model, các biến đổi quả học và cho ra mẫu dữ liệu, quy luật hoặc bộ công thức, những cái được gọi là kiến thức.

2.1.2 Quy trình khai phá tri thức

Hình dưới đây mô tả đầy đủ về quy trình khai phá tri thức.



Hình 2.1: Quy trình khai phá tri thức.

Như đã nói ở trên Data Mining là một phần của quá trình này. Tất cả các dữ liệu hiện có đều phải qua quy trình này để tạo nên nguồn tri thức.

Quy trình trên gồm 6 bước:

1. Làm sạch và tích hợp dữ liệu: Ở bước này ta loại bỏ nhiễu và dữ liệu không nhất quán, kết hợp với đó ta kết hợp nhiều nguồn dữ liệu có liên quan với nhau để có được tập dữ liệu đầy đủ thông tin nhất, bước này còn được gọi là làm sạch dữ liệu. Dữ liệu đầu ra sẽ được lưu trữ trong kho dữ liệu.
2. Chọn dữ liệu: Chọn các cột hay thuộc tính có liên quan đến nhiệm vụ cần xử lý từ kho dữ liệu.
3. Chuyển đổi dữ liệu: Ở bước này dữ liệu được chuyển đổi, mã hóa và hợp nhất để phù hợp với các quá trình tính toán.
4. Data Mining: Bước quan trọng nhất, ở đây sử dụng các phương pháp thông minh để trích xuất các công thức, mẫu và tri thức…
5. Đánh giá mẫu: Xác định độ hiệu quả, tính ứng dụng của kiến thức được rút ra từ quá trình Mining.
6. Trình bày kiến thức: Kiến thức được trình bày với lãnh đạo để đưa ra quyết định hoặc thể hiện dưới dạng đồ họa cho người dùng trải nghiệm.

2.1.3 Những loại Data và cách thức Data Mining

2.1.3.1 Những loai Data có thể khai phá.

Data Mining được coi là một phương pháp chung nhất trong công nghệ, vì vậy nó được dùng cho bất cứ loại dữ liệu nào kể cả multi media miễn là nó có tác dụng đối với mục đích đề ra. Đa số các loại dữ liệu được đưa vào phân tích hiện nay bao gồm dữ liệu từ kho dữ liệu, dữ liệu dưới dạng bảng ghi và dữ liệu giao dịch. Ngoài ra đang dần phát triển các dạng dữ liệu như hình ảnh, âm thanh, chuỗi thời gian, đồ thị, văn bản… Các công cụ gần đây đang tập trung phân tích các dạng dữ liệu đó. Trong các dữ liệu trên loại dữ liệu giao dịch và bảng ghi được sử dụng rộng rãi trong việc quyết định chiến lược và quản lý của danh nghiệp, mặt khác dữ liệu âm thanh, hình ảnh, văn bản thường được ứng dụng nhiều hơn trong đời sống hàng ngày. Những dữ liệu dạng đặc biệt này thường khó khai phá hơn nhưng bù lại nó có nhiều kiến thức được chứa trong đó và có tính ứng dụng rộng rãi hơn.

2.1.3.2 Những cách thức khai phá dữ liệu.

Ở mục 2.1.3.1 chúng ta đã thấy có rất nhiều loại dữ liệu được sử dụng, ở phần này ta sẽ tìm hiểu sơ lược về các cách thức mà ta có thể thực hiện với những loại dữ liệu đó. Nhưng nhìn chung chúng được chia làm 2 loại là phân loại và mô tả dữ liệu nói chung và 4 loại bao gồm khai phá luật kết hợp, phân lớp và hồi quy, gom cụm và xử lý ngoại lệ. một số trường hợp phương pháp xử lý ngoại lệ được kết hợp với tiền xử lý dữ liệu.

2.2 Tiền xử lý dữ liệu

Ta đang ở trong thời đại với một lượng khổng lồ data được nhận mỗi ngày, hiển nhiên sẽ có những data có ích và ngược lại. Thế nào là data không có ích, data không có ích là những data không liên quan đến mục đích ta cần đạt được, data bị nhiễu, bị thiếu, không nhất quán và đồng thời cũng không tin cậy. Điều này được coi là rất bình thường trong một thời đại được coi là "hôn loạn" của nguồn thông tin. Dữ liệu với chất lượng thấp hay không có ích sẽ để lại ảnh hưởng rất lớn cho quá trình khai phá tri thức. Vậy vấn đề đặt ra là: Làm thế nào để dữ liệu trở nên "sạch sẽ" và có chất lượng, Và ta thực hiện chúng như thế nào?

2.2.1 Thuộc tính của dữ liệu.

Thuộc tính dữ liệu đại diện cho các cột của bộ dữ liệu, thường được gọi một số tên khác như feature, dimention, variable. Nhiều thuộc tính miêu tả 1 đối tượng gọi là 1 vector. Trong giai đoạn chọn lọc dữ liệu ta thường chọn ra các thuộc tính liên quan và đồng nhất nhất. Càng nhiều thuộc tính thì đối tượng được mô tả càng cụ thể, tính chính xác của thuật toán càng cao nhưng đổi lại thời gian huấn luyện lâu hơn và phức tạp hơn.

Một số loại thuộc tính:

Thuộc tính danh nghĩa: thường chỉ ra tên, trạng thái, mã số hoặc danh mục của đối tượng. Ngoại trừ tên ra thì các loại khác của thuộc tính danh nghĩa thường được mã hóa sang thuộc tính binary hoặc thuộc tính thứ tự.

Thuộc tính binary: đơn giản với 2 trạng thái là 0 và 1. Ta thường định nghĩa bit 1 với những dữ liệu hiến gặp hơn hoặc mang nghĩa tích cực và ngược lại với bit 0.

Thuộc tính thứ tự: là loại thuộc tính dùng để đánh giá mức độ, thứ tự hoặc đánh dấu các danh mục.

Thuộc tính số: Là thuộc tính định lượng, nghĩa là thuộc tính này được rút ra từ việc đo lường và được biểu diễn bằng số.

2.2.2 Tại sao phải tiền xử lý dữ liệu.

Ta đã nói tóm tắt về chất lượng của dữ liệu, dữ liệu chất lượng là dữ liệu có thể đưa vào huấn luyện, đáp ứng đúng và đủ mục đích sử dụng đặt ra. Thường dữ liệu được thu thập từ người dùng hằng ngày, nhưng vì đây là con người nên thông tin được cung cấp sẽ không cung cấp, không đúng định dạng và thậm chí bỏ trống. Chắc chắn, với nguồn cung cấp "lỗi lõm" như vậy ta không thể rút ra được nguồn tri thức đáng tin cậy. Những lỗi đó bắt nguồn từ nhiều lý do mà chúng ta không thể can thiệp và không thể giải quyết, ví dụ: lỗi nhập liệu người dùng, sai định dạng từ phần mềm thu thập, sự khác nhau về phiên bản, phần mềm…Tưởng tượng một đầu bếp không thể tự bắt con cá anh ta muốn để chế biến món ăn anh ta thích mà anh ta phải tự chế biến lại từ những thứ anh ta nhận được. Mỗi mục đích khác nhau cần chất lượng dữ liệu khác nhau. Ngoài ra tính "mới nhất" cũng cần thiết đối với dữ liệu, không thể dùng dữ liệu bán hàng từ 2 năm trước để dùng cho hiện tại.

2.2.3 Các nhiệm vụ chính trong tiền xử lý dữ liệu.

2.2.3.1 Làm sạch dữ liệu

Dữ liệu trong thế giới thực có xu hướng không đầy đủ và không nhất quán. Với các trường hợp thiếu dữ liệu, có nhiều cách: Xóa thuộc tính bị thiếu, thay thế bằng giá trị trung bình, giá trị phổ biến nhất, đặt cho tất cả giá trị thiếu bằng 1 biến (Ví dụ: "Missing").

2.2.3.2 Giảm dữ liệu

Giảm dữ liệu bao gồm giảm kích thước, giảm số lượng. Các phương pháp giảm kích thước bao gồm wavelet transforms, PCA hoặc loại bỏ các thuộc tính không liên quan. Các phương pháp giảm số lượng áp dụng các mô hình hồi quy dùng tham số để

ước tính lượng dữ liệu ngoài ra còn có các hương pháp phi tham số khác như gom cụm, lấy mẫu để chọn ra lượng dữ liệu cần thiết.

2.2.3.3 Chuyển đổi dữ liệu

Trong chuyển đổi, dữ liệu được chuyển đói thành một hình thức phù hợp để khai thác. Bao gồm các hoạt dộng như tổng hợp, chuẩn hóa, rời rạc hóa và chuyển đổi shape của dữ liệu. Trong các loại trên, chuẩn hóa được coi là quan trọng nhất.

2.3 Luật kết hợp

2.3.1 Khái niệm.

Khai thác luật kết hợp là khai thác, tìm kiếm các mối quan hệ thường xuyên xuất hiện trong một tập dữ liệu. Mục tiêu ban đầu là phân tích nguồn dữ liệu bán hàng để đưa ra các xu hướng người dùng nhưng đã được sử dụng trong nhiều lĩnh vực khác nhau. Việc khai thác các mối quan hệ tương quan giúp ích rất nhiều trong việc đưa ra quyết định kinh doanh. Marketing hay phân tích hành vi khách hàng. Ví dụ: Sau khi khám mắt khách thường có xu hướng mua kính hoặc sau khi mua sữa khách lại mua thêm bánh mì…

Trong một bộ dữ liệu có các đối tượng cấu tạo thành, ví dụ như dữ liệu bán hàng sẽ có các đối tượng như sữa, bánh mì, nước uống v.v. Các đối tượng này sẽ kết hợp tạo ra các luật dạng X→Y với X, Y thuộc bộ đối tượng và X giao Y = rỗng

Ví dụ: {Trứng, Sữa}→ {Bánh mì} Người mua trứng và sữa sẽ mua bánh mì.

Itemset: Là tập gồm 1 hoặc nhiều sản phẩm. Ví dụ: {Trứng, Sữa}

k-itemset là itemset có k phần tử.

Support count (σ): Tần suất xuất hiện của 1 itemset

Support: Tỉ lệ xuất hiện của itemset trong tất cả giao dịch

Frequent itemset:Là itemset có support >= ngưỡng **minsup**

Với 1 luật X→Y ta có 2 giá trị cần tính là Support (X, Y) và Confidence (c)

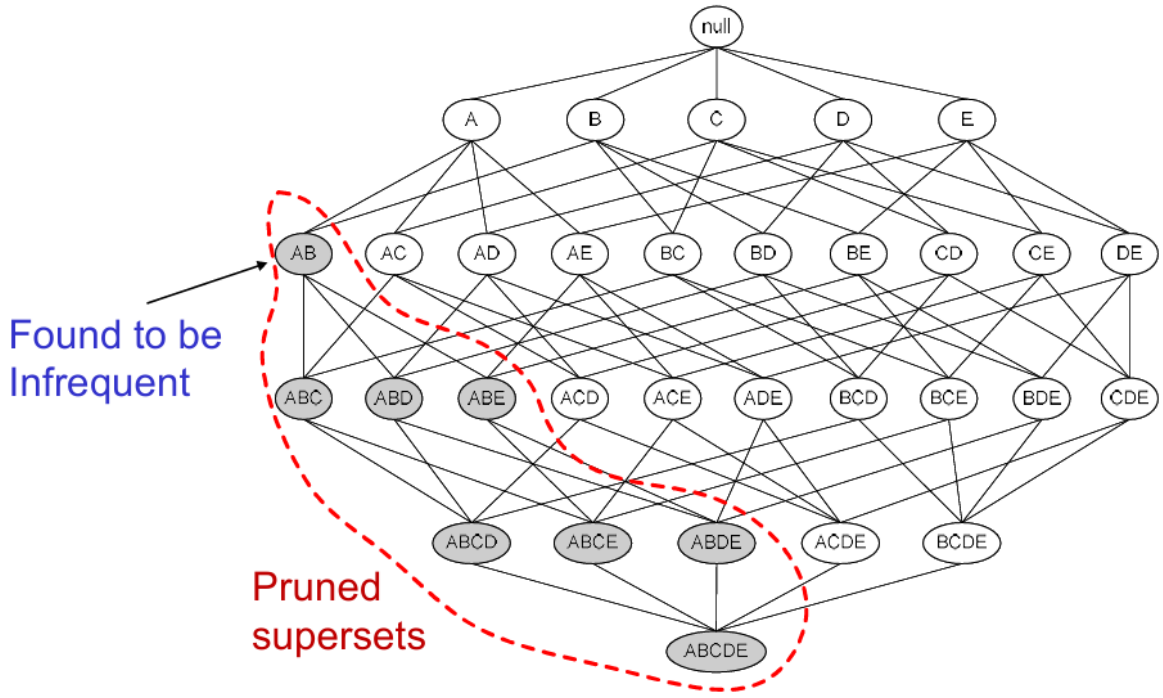
Với Confidence = Support count (X, Y)/Support count (X)

2.3.2 *Cách xây dựng một bộ Rule*

Thực hiện theo các bước sau:

1. Tìm tất cả các tập frequent itemset.
2. Sử dụng tập đó để xây dựng các luật

Cách thông thường là liệt kê ra tất cả các itemset và tính support cho từng itemsets. Chọn ra các frequent itemsets và từ đó xây dựng các luật. Chọn ra các luật trên ngưỡng conf



Hình 2.2: Mô tả cách liệt kê các item set.

Nhược điểm của cách thông thường này là có quá nhiều itemsets được đưa ra và tốn rất nhiều chi phí tính toán. Một phương pháp khác có thể giải quyết vấn đề này là APRIORI.

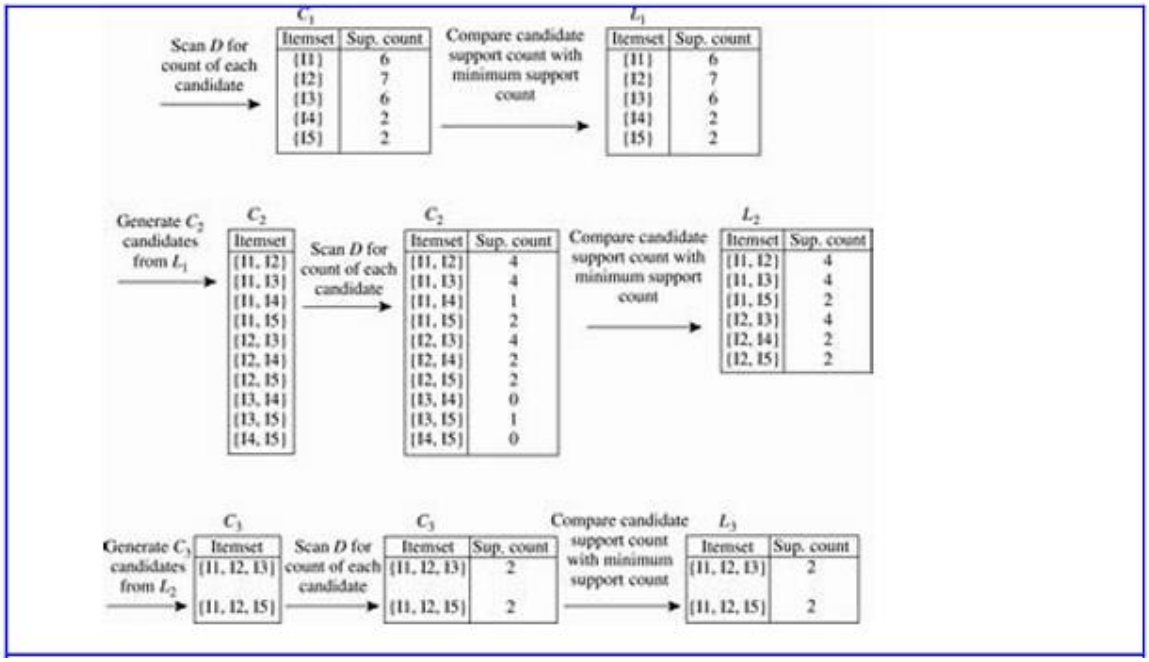
2.3.3 *Phương pháp Apriori.*

Cách tiếp cận của phương pháp này là tìm tập frequent của k-itemsets và từ đó suy ra tập frequent của tập k+1 – itemsets. Đến đây nhìn chung thuật toán vẫn còn khá phức tạp khi phải tìm các tập frequent, vì vậy trong phương pháp này có thêm các thuộc tính để giảm không gian tìm kiếm.”Nếu 1 itemset là frequent thì tất cả tập

con của nó là frequent" từ phát biểu này ta chỉ cần tìm các tập frequent có k lớn nhất và suy ra các tập con. Từ phát biểu 1 cũng suy ra thêm "nếu 1 itemset có không là frequent thì tập chứa nó cũng không là frequent", tính chất này dùng để giảm không gian tìm kiếm rất hiệu quả, giả dử A không frequent thì các tập chứa A cũng không frequent.

Các bước thực hiện được tiến hành như sau:

1. Tìm tất cả 1-items set, lấy những frequent items set, từ những frequent items set lấy được tạo các tập 2 – items set, cứ thế tiếp tục đến k – item set với số lượng frequent k – item set =1
2. Từ các frequent items set nhận được ta xây dựng các luật. Ví dụ ta nhận được tập {A, B, C} ta sẽ xây dựng được các tập {A, B}→ C {A, C}→ B {B, C}→ A {A}→{B, C} {B}→{A, C} ….
3. Từ các luật tìm được, tính confidence và đưa ra kết quả.



Hình 2.3: Minh họa kết quả sau các bước lựa chọn frequent itm set.

Như hình trên ta đã rút ra được các frequent items set L1, L2, L3 với support cụ thể. Từ L1, L2, L3 ta xây dựng các luật, thường thì ta đi tìm luật từ frequent k – items set với k lớn nhất sau đó giảm dần để tránh trùng.

2.3.4 Tổng hợp.

Việc tiến hành tìm các luật kết hợp, các mẫu rất hữu ích trong mục đích kinh doanh và phân tích hành vi khách hàng từ các dữ liệu giỏ hàng và hóa đơn mua bán.

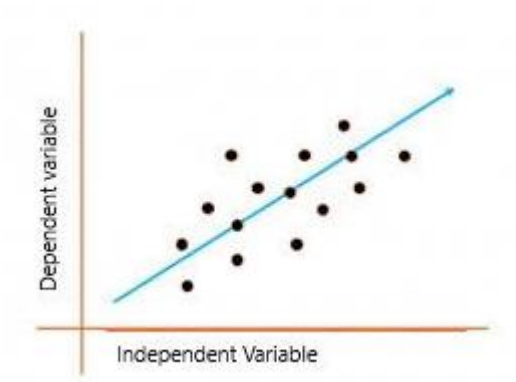
Khai phá các luật kết hợp là tìm các mối quan hệ {A}→{B} diễn ra thường xuyên và đáp ứng được độ tin cậy đưa ra.

2.4 Phân lớp và dự đoán

2.4.1 Dự đoán (Linear regression).

Bìa toán dự đoán hay còn gọi là linear regression là một thuật toán mang tính nền tả trong ứng dụng dữ liệu. Đến nay nó vẫn được ứng dụng rộng rãi với nhiều ứng dụng trong nhiều lĩnh vực.

Sơ lược về bài toán, cho một tập dữ liệu, ở quá trình training thuật toán cố gắn tìm kiếm 1 đường thẳng sao cho độ chênh lệch giữ đường thẳng và các điểm trong training data set là thấp nhất. Dùng đường thẳng đó để đoán các giá trị mới với đầu vào là x, phương trình đường thẳng y có dạng y =w.x+b, với các tham số w, b là các giá trị mà model được học.



Hình 2.4: Ví dụ về Linear Regression.

Để tính toán sự chênh lệch của thuật toán dự đoán ta có một số phương pháp MSE, MRE, R-square.

Hồi quy tuyến tính tuy là thuật toán đơn giản nhất nhưng nó được coi là nền tảng cho các kỹ thuật khác như logistic regression, neural networks…Ngoài ra bài toán dự đoán còn được sử dụng trong nhiều lĩnh vực như tài chính, kinh tế, y tế và cả dự báo thời tiết.

2.4.2 Phân lớp (Classification)

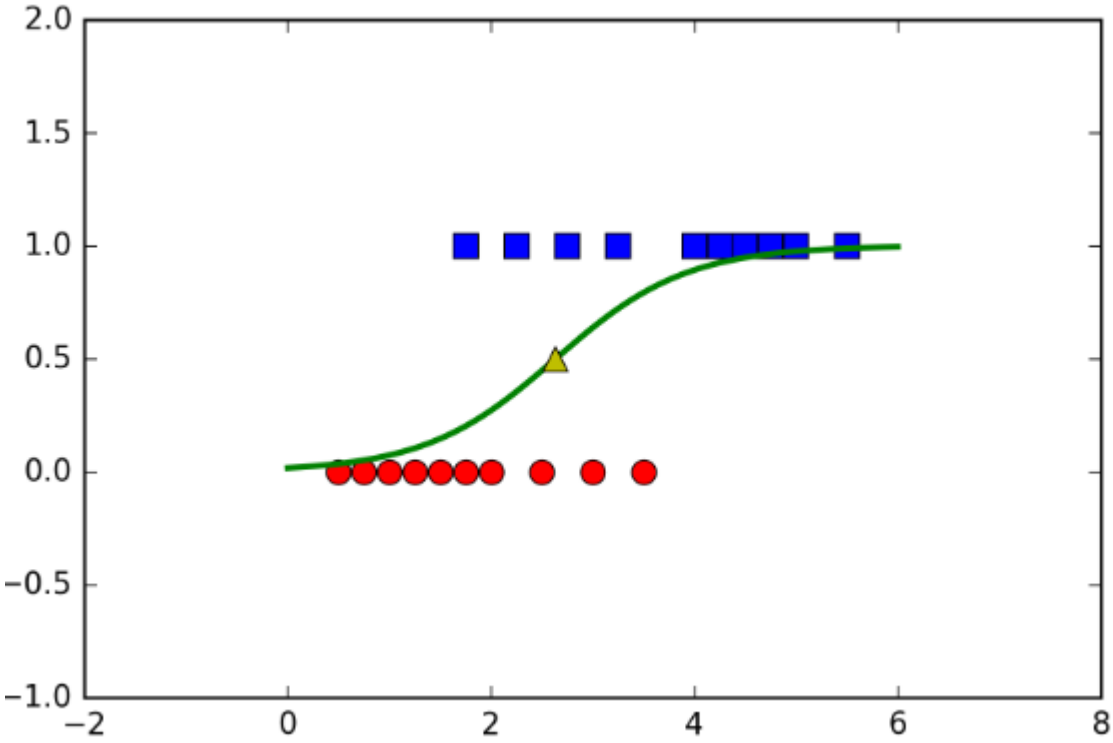
2.4.2.1 Khái niệm.

Là một loại bái toán trong phân tích dữ liệu, dựa vào dữ liệu train để huấn luyện mô hình và dùng mô hình để dự đoán, khác với linear regression cho đầu ra là các số

liên tục mà ta không biết trước thì bài toán classification có đầu ra cố định, rời rạc với các giá trị là các category được chỉ định từ đầu. Ví dụ: đưa vào 100 tin nhắn email và phân lớp email nào có nội dung spam và không spam.

2.4.2.2 Các dang bài toán phân lớp.

1. Logistic regression: là dạng bài toán đơn giản nhất của classification với đầu vào A và đầu ra là 1 trong 2 giá trị 1 và 0. Bài toán có 1 hàm non-linear (sigmoid, tanh, …) ở đây ta chọn sigmoid, đầu vào A được biến dổi linear A= w.x+b. Ta có y = sigmoid (A) là một biểu đồ non-linear với giới hạn theo trục y từ 0 đến 1. Với các giá trị x cho ra y >=0.5 được phân vào lớp 1 và ngược lại được cho vào lớp 0.



Hình 2.5: Mô tả hình dạng của hàm Sigmoid.

2. Naïve Bayes Classifier: Cách nhìn chung của bài toán này là cho các thuộc tính của x, tính sác xuất để x thuộc vào lớp c, và t chọn lớp có sác xuất cao nhất để làm đại diện cho bộ x đó. Viết gọn là p(c|x). mục tiêu bài toán là tìm sát xuất p cao nhất.
3. Decision Tree: là một dạng bài toán khác của classification. Cây là một tập hợp các câu hỏi với các câu trả lời, cả 2 đều được gọi là các node. Các câu

hỏi có thứ tự trước sau, câu trả lời của câu hỏi trên sẽ dẫn đến câu hỏi bên dưới. Từ các câu hỏi và câu trả lời ta sẽ phân loại được dữ liệu. Dựa vào dữ liệu huấn luyện ta sẽ xác định được các câu hỏi và thứ tự của chúng.

2.4.2.3 Naïve Bayes Classifier.

Nhắc lại ở trên, cho một đầu vào x và giả thiết x thuộc c. Ta có bộ sát xuất p(c|x), nhiệm vụ cần làm là từ tập dữ liệu train rút ra các tham số để dự đoán các sát xuất cho từng class và rút ra class có sát xuất cao nhất.

Vậy sác xuất p(c|x) được đính như thế nào? P(c|x)=[p(c)*p(x|c)]/p(x) (1)

Với p(x|c)= p(x1, x2, …, xn|c) và p(x) = tổng[p(x|ci)] chú ý: ci ở đây chỉ tất cả các class trong C (2)

Trong công thức số 1 kết hợp với 2, điều duy nhất ta cần tìm là p(x|c)

Ở đây ta chỉ xét các dữ liệu có giá trị liên tục nên công thức của p(xi|c) được tính như sau:

$$p(x_i|c) = p(x_i|\mu_{ci}, \sigma_{ci}^2) = \frac{1}{\sqrt{2\pi\sigma_{ci}^2}} \exp\left(-\frac{(x_i - \mu_{ci})^2}{2\sigma_{ci}^2}\right)$$

Trong đó, bộ tham số $\theta = \{\mu_{ci}, \sigma_{ci}^2\}$ được xác định bằng Maximum Likelihood:

$$(\mu_{ci}, \sigma_{ci}^2) = \arg\max_{\mu_{ci}, \sigma_{ci}^2} \prod_{n=1}^{N} p(x_i^{(n)}|\mu_{ci}, \sigma_{ci}^2) \quad (9)$$

Có thể tính 2 tham số kia như sau:

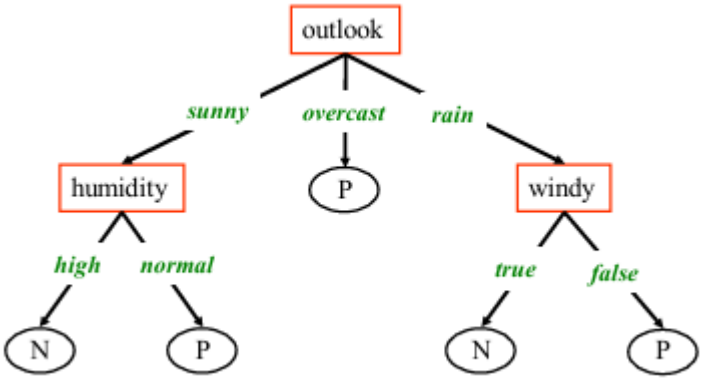
Muy i của class c = #exp in class c/tổng số class = c (A)

Sigma i của class c = căn(tổng(từng xi – muy i c)/ số hàng) (B)

Như vậy, khi nhận 1 hàng dữ liệu mới, ta tính p(x|c) của từng thuộc tính theo từng class bằng cách thế các tham số (A) và (B) được tính từ trước, sau đó đem thế vào công thức (1) theo từng class và chọn class có sát xuất cao nhất

2.4.2.4 Decision Tree.

Nhắc lại, Cho một bộ dữ liệu X chưa xác định nhãn, các thuộc tính của X được đưa vào Decision Tree và theo các chỉ dẫn từ cây đưa ra dự đoán cho bộ X đó. Decision Tree đơn giản vì nó không cần học bất kỳ một param nào, cách thể hiện kiến thức của nó thân thiện hơn so với các thuật toán khác. Cách thức hoạt động của Decision Tree giống như một bộ quy tắc.



Hình 2.6: Một mô hình cây quyết định.

Ở đây ta sẽ tìm hiểu về một loại thuật toán rất phổ biến đó là ID3. Đây là một bài toán được sủ dụng cho dạng dữ liệu category. Hướng đi của bài toán này là ta cần phải xác định thuộc tính tốt nhất ở mỗi bước dựa trên một tiêu chuẩn nào đó. Các bước ở đây được xem như các câu hỏi, sau mỗi câu hỏi dữ liệu được phân chia vào các child node. Để đánh giá độ chính xác trong phân loại, ta đặt ra một ngưỡng, nếu dữ liệu trong các child node vẫn còn lẫn vào nhau (đa số không cùng 1 class) thì là phép phân chia chưa thực sự tốt và ngược lại. vì vậy là cần một bộ số đo độ tinh khiết (purity) hoặc độ vẫn đục (impurity). Hàm số để tính các giá trị này là hàm Entropy.

Hàm Entropy có công thức như sau:

$$H(\mathbf{p}) = -\sum_{i=1}^{n} p_i \log(p_i) \qquad (1)$$

Với pi là phân phối sát xuất của mỗi giá trị trong 1 thuộc tính (cột).

Entropy càng thấp tức mức độ tinh khiết của p càng cao.

Cụ thể với ID3. **Xét bài toán với C class (C output – giải đoạn mở đầu)**, xét trên 1 node nào đó là non leaf node, trên node đó có N phần tử (hàng), trong N phần tử đó sẽ có Nc phần tử thuộc class c với c thuộc C, ta có công thức tính Entropy tại node đó:

$$H(S) = -\sum_{c=1}^{C} \frac{N_c}{N} \log\left(\frac{N_c}{N}\right) \qquad (2)$$

Giả sử chọn thuộc tính x, với mỗi thuộc tính sẽ có K giá trị unique, ta sẽ có K child node S1, …, Sk và số điểm trong mỗi child node này là m1, …, mk. Ta có công thức:

$$H(x, S) = \sum_{k=1}^{K} \frac{m_k}{N} H(S_k) \qquad (3)$$

Tiếp theo ta có thêm một định nghĩa là information gain của x:

$$G(x, S) = H(S) - H(x, S)$$

Giá trị G này giúp xác định thuộc tính nào được chọn ở node hiện tại, thuộc tính được chọn là thuộc tính làm cho giá trị của G lớn nhất. tương đương với H(x, S) nhỏ nhất. Khi chọn được thuộc tính này sẽ có K child node được tạo kèm theo. Dựa vào H(Sk) so sánh với ngưỡng tinh khiết để lựa chọn các thuộc tính khác (lúc này ta lại xét bài toán C class với C là số class của thuộc tính x).

Thuật toán phân chia này cần điều kiện dừng:

1. Nếu node cón Entropy = 0 tức là nó tinh khiết.
2. Số phần tử của node nhỏ hơn một ngưỡng nào đó thì dừng.
3. Cây đạt đến một số lượng ‘Tầng” nhất định sẽ dừng.
4. Việc phân chia không làm giảm Entropy quá nhiều.

2.4.2.5 Đánh giá mô hình.

Dữ liệu thường được chia thành tập test và train, ta thường kiểm tra độ chính xác của thuật toán trên tập test. Với 1 data set lớn thường có nhiều cách chia tập train-test. Cách thông thường đơn giản nhất là chia theo tỉ lệ train-test là 7-3 hoặc 8-2.

Nhưng với các trường hợp với số dữ liệu nhỏ hoặc người lập trình muốn tối ưu hóa các parameter thì train và test được chia theo phương pháp k fold cross validation. Phương pháp này chia data set thành k folds nhỏ hơn, lấy 1 folds làm test và k-1 folds còn lại làm train, lặp lại quá trình với các folds còn lại. Mô hình này rất hữu ích với data có số lượng nhỏ hoặc khó tìm kiếm, ví dụ: với dữ liệu 1000 mẫu máu với phương pháp thông thường khi ta lấy 700-800 mẫu để train và chỉ có 300 -200 điểm để test gây ra lãng phí dữ liệu.

Tiếp theo với phần đánh giá cho bài toán classification ta dùng confusion matrix.

Actual class\Predicted class	C_1	$\neg C_1$
C_1	**True Positives (TP)**	**False Negatives (FN)**
$\neg C_1$	**False Positives (FP)**	**True Negatives (TN)**

Hình 2.7: Confusion Matrix.

Ta tính độ chính xác của thuật toán bằng công thức:

Accuracy = (TP, TN)/All

Nhưng lại gặp một số vấn đề, một số lớp trong dữ liệu có thể hiếm gặp, ví dụ có 5 lần xuất hiện class hiếm 0 nhưng ta chỉ dự đoán đúng được 2, với công thức bên trên thì độc hính xác đưa ra sẽ không phản ánh đúng, đây được gọi là vấn đề mất cân bằng. Để giải quyết vấn đề trên ta đưa ra các khái niệm khác là recall, precision và F1-Score với:

Precision = TP/(TP+FP)

Recall = TP/(TP+FN)

F1-Score = (2 x Precision x Recall) / (Precision + Recall)

Phương pháp này giúp ta cân bằng được các lớp. Khi một lớp hiếm hơn được phân lớp sai nhiều thì ảnh hưởng của nó trong độ chính xác sẽ lớn hơn so với cách tính accuracy thông thường.

2.5 Gom cụm

2.5.1 Khái niệm.

Là bài toán học không giám sát, một dạng khác của classification. Thay vì ta biết trước các class để phân đối tượng vào thì ở bài toán phân cụm ta không được biết trước điều đó, mà từ dữ liệu, bài toán tự tìm ra các tập hợp của dữ liệu hay còn gọi các cụm, các dữ liệu trong 1 cụm tương tự nhau và khác biệt so với cụm khác, đại diện mỗi cụm là 1 center, khi cho một đối tượng mới vào dựa vào sự tương đồng của đói tượng mới và các điểm trung tâm mà phân nó vào cụm tương ứng. Ngoài ra phân cụm còn giúp phân chia dữ liệu, tìm ra hương pháp quản lý cụ thể cho từng loại đối tượng.

2.5.2 Ứng dụng.

Phân cụm được sử dụng rộng rãi trong nhiều ứng dụng như kinh doanh thông minh, nhận dạng hình ảnh, … Trong kinh doanh thông minh, phân cụm được dùng để phân chia lượng lớn khách hàng thành các nhóm, nơi mỗi nhóm có các hành vi mua hàng giống nhau. Điều này tạo điều kiện cho việc phát trieenr chiến lược kinh doanh và quản lý khách hàng. Trong phân tích hình ảnh, lấy ví dụ chữ viết tay, ta có thể gom cụm những chữ viết tay của cùng 1 ký tự vào 1 cụm bằng cách xem mỗi hình ảnh viết tay là 1 điểm dữ liệu, ứng dụng trong cả tách vật thể, từ 1 hình ảnh bình thường có k màu chủ đạo, ta có thể tách ảnh ra thành k cụm, ví dụ cụm 1 có màu đen là tóc, cụm 2 có màu trắng là răng …

2.5.3 Thuật toán gom cụm.

Mục tiêu của thuật toán gom cụm là tìm ra trumg tâm của các cụm. Ở đây ta sử dụng thuât toán đơn giản nhất là k-mean cluster. Mục tiêu tối ưu của bài toán là tìm các trung tâm sao cho khoảng cách từ các điểm trong cụm đến trung tâm của cụm đó là nhỏ nhất.

Ta sẽ tiếp cận bài toán bằng 2 hoạt động sau, cố định trung tâm tìm cụm sau đó cố định cụm tìm trung tâm. Cả 2 quá trình này sẽ được lặp lại cho đến khi mất mát là nhỏ nhất. Xét bộ dữ liệu có N điểm, dầu tiên ta chọn random k điểm làm trung tâm, với k điểm trung tâm cố định, ta xét từng điểm trong N, điểm thành viên nào có khoảng cách đến điểm trung tâm của cụm tương ứng thấp nhất thì thuộc về cụm đó. Sau khi chạy hết N điểm dữ liệu thì ta có được k cụm tạm thời. Tiếp theo từ k cụm cố định đó, trong mỗi cụm ta lấy trung bình tọa độ của các điểm trong cụm cho ra tọa độ của điểm trung tâm mới, có nghĩa là từ k cụm cố định ta tìm được k điểm trung tâm mới. Với 3 điểm trung tâm mới ta lại tiếp tục phân chia các điểm N lại từ đầu, được các cụm mới. Cứ thế lặp lại cho đến khi sự sai khác của các điểm trung tâm không còn đáng kể thì ta đã hoàn thành phân cụm.

2.5.4 Những bài toán gom cụm mở rộng.

Thuật toán gom cụm bên trên có một vài hạn chế, chúng ta có thể không biết chúng phân cụm như thế nào nhưng cần phải biết cần phân ra bao nhiêu cụm, có một số phương pháp chọn số cụm như Elbow Method. Ngoài chọn số lượng cụm, vị trí ban đầu của các điểm trung tâm cũng quan trọng, Việc chọn các điểm trung tâm hợp lý sẽ giảm đáng kể thời gian huấn luyện của thuật toán. Số điểm dữ liệu trên mỗi cũm cũng ảnh hưởng, nếu có sự chênh lệch quá lớn trong số lượng điểm trên mỗi cụm, điều này gấy ra kết ủa không chính xác.

Phương pháp được để cập ở trên chỉ áp dụng hiệu quả với các cụm có dạng hình tròn, những dữ liệu có dạng dẹt hoặc 1 cluster nằm trong một cluster sẽ gây ra gom cụm sai. Để giải quyết vấn đề này ta có các giải thuật gom cụm thay thế khác như: DBSCAN, OPTICS, DENCLUE…

2.6 Xu hướng của Data Mining

Trong một thời đại nở rộ về dữ liệu, khai thác dữ liệu dần trở nên là một công cụ không thể thiếu trong loạt các lĩnh vực. Tuy phát triển nhanh và ứng dụng rộng rãi nhưng data mining vẫn còn có những thách thức đối với những loại dữ liệu phức tạp, cách các mô hình xử lý các dạng dữ liệu này thực sự chưa đạt được mức tin tưởng từ con người.

Các dữ liệu phức tạp mà hiện nay đang được đưa vào khai thác nhiều hơn là các dữ liệu dạng chuỗi: time-series, chuỗi biểu tượng, chuỗi sinh học. Là một chuỗi thông tin và sự kiện được sắp xếp có thứ tự.

Trong dữ liệu time-series bao gồm chuỗi dữ liệu số được ghi lại sau một khoảng thời gian bằng nhau, những dữ liệu này thường được tạo ra từ các quá trình tự nhiên như thời tiết, kinh tế thị trường như chứng khoáng, giá vàng hay các quá trình phát triển tế bào trong y tế.

Đối với chuỗi biểu tương, nó là một tập hợp các yếu tố, hoạt động có trật tự nhưng không đề cập đến các khoảng thời gian, ví dụ như chuỗi hành động mua hàng của khách, chuỗi các thao tác click chuột hoặc chuỗi thực thi các chương trình.

Chuỗi sinh học, loại dữ liệu thường thấy trong lĩnh vực y tế, thường là chuỗi DNA hoặc chuỗi Protein. Có vai trò quan trọng trong phân tích trình tự hay sinh ra những loại mới, nhận dạng, lai hóa thậm chí tiến hóa của sinh vật.

Ngoài các dữ liệu dạng chuỗi còn các dữ liệu bán cấu trúc và phi cấu trúc khác như dữ liệu không gian, multi media, siêu văn bản cũng có nhiều ứng dụng. Dữ liệu này mang nhiều ngữ nghĩa và thông tin hơn và xuất hiện rộng rãi hơn trong cuộc sống. Gần gũi nhất vẫn là dữ liệu multi media bao gồm âm thanh, hình ảnh, video. Khai thác những dự liệu này là tích hợp của nhiều nhiệm vụ như xử lý hình ảnh, thị giác máy tính, nhận diện giọng nói … các vấn đề trong khai thác loại dữ lieuj này là truy xuất dựa vào nội dung, tìm kiếm, phân tích đa chiều cũng như dự đoán và phân lớp. Với dữ liệu là text, đây là lĩnh vực đang được chú trọng nhiều nhất hiện nay. Có thể xem tầm quan trọng của nó giống như tầm quan trọng của biết chữ đối với con người và ứng dụng của phân tích text data là tất cả những điều mà con người thường

hay làm như đọc email, phân tích cảm xúc, dịch thuật, trả lời câu hỏi, kể chuyện… Một mô hình khai thác text data được coi là chất lượng nếu nó có sự kết hợp chặt chẽ, mới lạ và thú vị giống như một bài văn vừa xuất bản của một tác giả nào đó.

Hiện nay, khai thác dữ liệu là một phần của cuộc sống hằng ngày, nó phổ biến đến mức ta còn không nhận ra sự hiện diện của nó ở thứ đơn giản nhất. mặc dù vậy nhưng vẫn có một số vấn đề, nó đòi hỏi mọi thứ trở nên nhanh hơn, liên tục đổi mới, lạm dụng sẽ mất khả năng sáng tạo và đặt biệt là mối lo ngại về an toàn bảo mật nguồn thông tin.

Chương 3. PHÂN TÍCH XÂY DỰNG MÔ HÌNH

3.1 Phân tích.

Dự báo thời tiết là một trong những lĩnh vực ứng dụng data mining rộng rãi và hiệu quả nhất. Với tính chất thời tiết tại 1 thời điểm sẽ phụ thuộc vào thời điểm trước đó, bộ dữ liệu chuỗi thời gian bao gồm các thuộc tính của thời tiết cùng với các mô hình sử dụng các trạng thái của các time-step trước để dự đoán kết quả ở hiện tại như RNN và LSTM là lựa chọn hợp lý cho bài toán dự đoán dự báo thời tiết.

3.2 Tiền xử lý dữ liệu.

Do tính chất phân tích của môn học và sự thực tế, bộ dữ liệu thời tiết của Thành Phố Hồ Chí Minh được lấy từ trang web visualcrossing.com. Để dễ dàng trong nhận xét và huấn luyện thuật toán, tránh lãng phí thời gian. Bộ train data bao gồm 12 tháng từ 4/2023 – 4/2024 và bộ test data là dữ liệu từ tháng 5/2024 đến 2/6/2024.

Vì các thuộc tính là các yếu tố tự nhiên nên tập data trên sẽ không có các trường hợp nan hoặc không đồng nhất. Và bộ dữ liệu này cũng đã có sẵn thứ tự theo thời gian.

Bộ dữ liệu thu được bao gồm các thuộc tính:

```
df.columns
```

```
Index(['name', 'datetime', 'temp', 'feelslike', 'dew', 'humidity', 'precip',
       'precipprob', 'preciptype', 'snow', 'snowdepth', 'windgust',
       'windspeed', 'winddir', 'sealevelpressure', 'cloudcover', 'visibility',
       'solarradiation', 'solarenergy', 'uvindex', 'severerisk', 'conditions',
       'icon', 'stations', 'month', 'hour'],
```

Hình 3.1: Các thuộc tính gốc của dữ liệu.

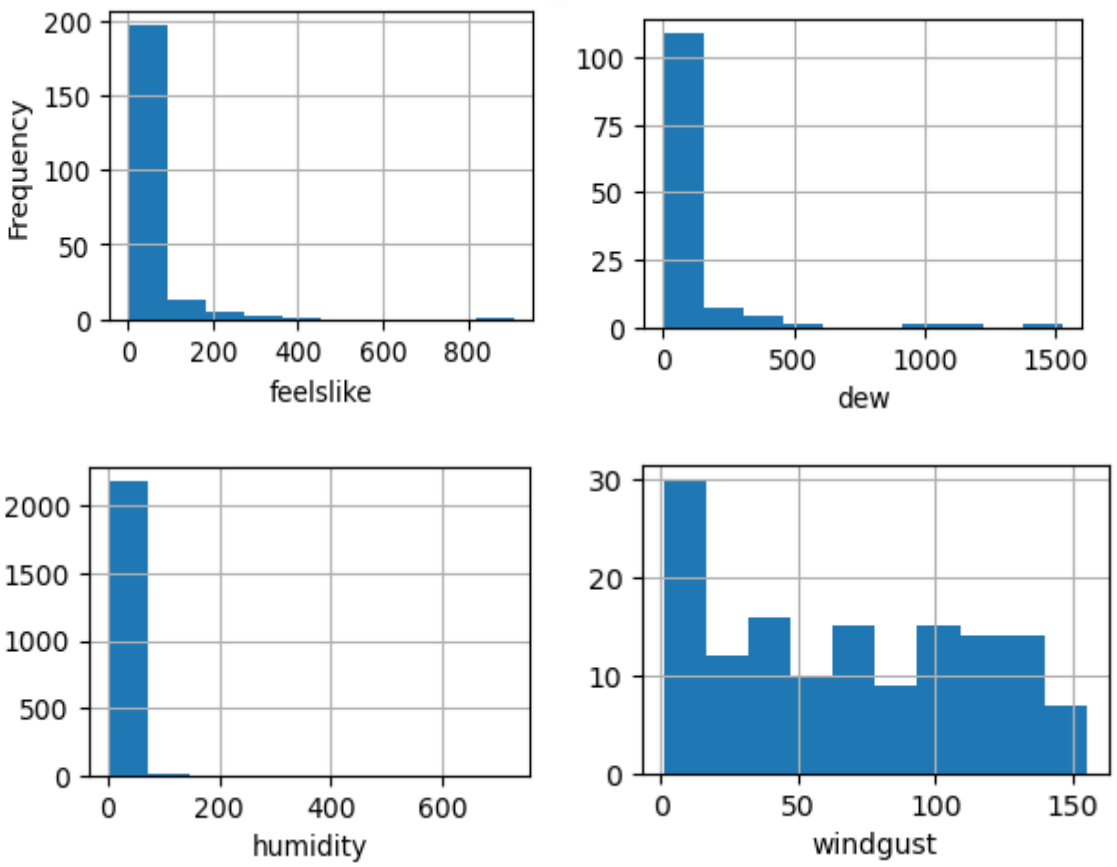
Sau khi thực hiện câu lệnh df.nunique(), ta loại bỏ những thuộc tính không cần thiết và giữ lại các thuộc tính sau: feelslike, dew, humidity, windgust, winddir, solarradiation, visibility, cloudover, month, hour.

Ở đây ta thấy có thêm 2 thuộc tính đó là ‘month’ và ‘hour’. Nhận thấy thời tiết phụ thuộc vào tính chất mùa và mỗi giờ trong ngày có một mức nhiệt độ nhất định. Nên từ cột ‘datetime’ ta tách ra làm 2 thuộc tính:

```python
df['datetime'] = pd.to_datetime(df['datetime'], utc=True)
# Trích xuất các thành phần từ cột datetime
df['month'] = df['datetime'].dt.month
df['hour'] = df['datetime'].dt.hour
```

Hình 3.2: Trích xuất tháng và giờ từ cột datetime

Sau khi có được bộ dữ liệu, ta vẽ biểu đồ để xem phân phối của từng thuộc tính.



Hình 3.3: Phân phối dữ liệu khi chưa chuẩn hóa.

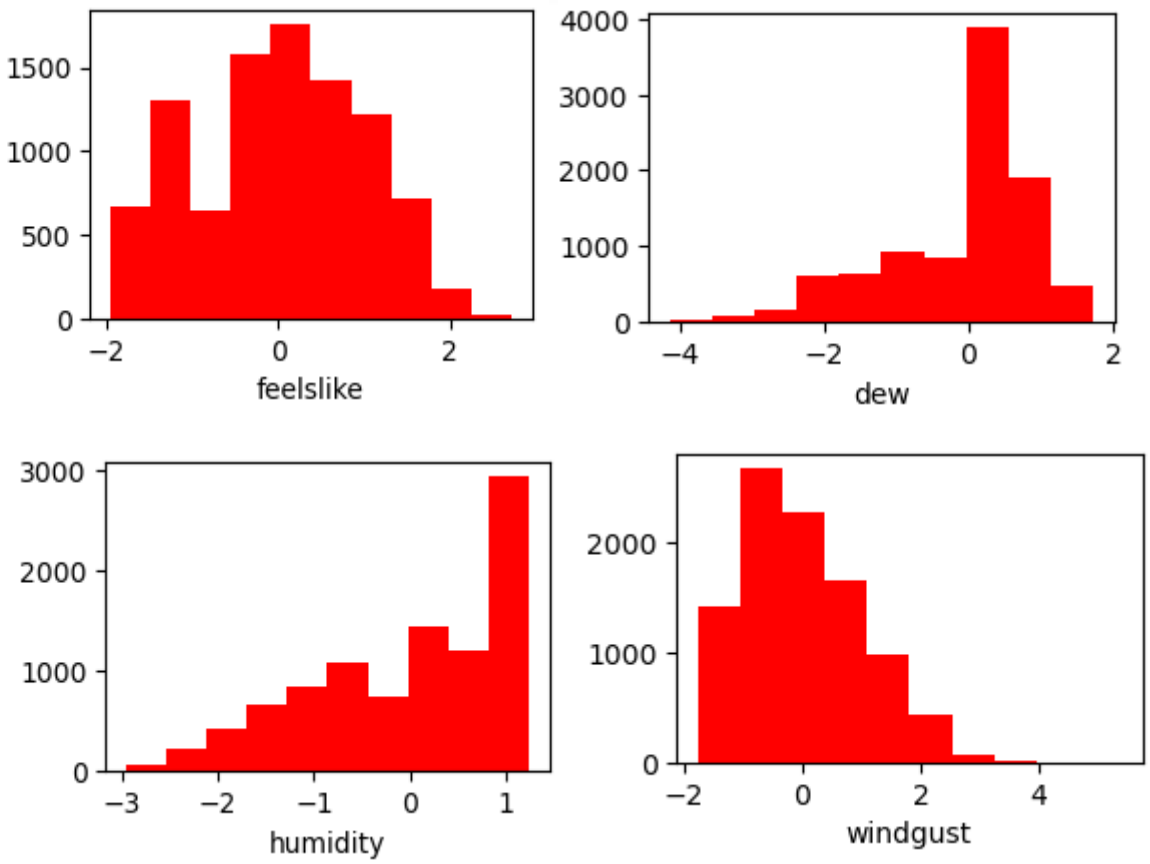
Vì các phân phối khá xấu nên ta phải tiến hành chuẩn hóa chúng. Ở đây chuẩn hóa Z-score (StandardScaler của sklearn) được sử dụng:

```python
from sklearn.preprocessing import StandardScaler, MinMaxScaler
```

```python
scalers = {}
for i, feature in enumerate(features):
    scalers[feature] = StandardScaler()
    X_train[:, i] = scalers[feature].fit_transform(X_train[:,i].reshape(-1, 1)).reshape(-1)
```

Hình 3.4: Chuẩn hóa với z-score.

Sau chuẩn hóa ta được các phân phối tương ứng như sau:



Hình 3.5: Dữ liệu sau khi chuẩn hóa.

Đã đẹp hơn rất nhiều, các giá trị được đưa về xung quanh gốc O(0; 0).

Yêu cầu dữ liệu đầu vào của LSTM và RNN theo bài toán là chuỗi n dữ liệu trước đó để dự đoán 1 điểm hiện tại nên ta sẽ chuyển dữ liệu từng dòng thành từng sequence với chiều dài bằng seq_length:

```
def create_sequences(X, y, seq_length):
    xs, ys = [], []
    for i in range(len(X) - seq_length):
        x_ = X[i:i+seq_length]
        y_ = y[i+seq_length]  # Dự đoán 1 lần nhiệt độ
        xs.append(x_)
        ys.append(y_)
    return np.array(xs), np.array(ys)

seq_length = 20 # Sử dụng 20 giờ để dự đoán giờ tiếp theo
X_seq, y_seq = create_sequences(X_train, y_train, seq_length)
```

Hình 3.6: Tạo sequence dữ liệu.

Để tận dụng GPU trong huấn luyện, chúng ta sẽ sử dụng thư viện Torch, vì vậy data phải được chuyển sang dạng tensor:

```
X_train = torch.tensor(X_seq, dtype=torch.float32)
y_train = torch.tensor(y_seq, dtype=torch.float32).unsqueeze(1)
```

Hình 3.7: Chuyển từ mảng Numpy sang Torch.

3.3 Đào tạo và kiểm tra mô hình

Cả LSTM và RNN đều được hỗ trợ trong thư viện Torch, ta chỉ cần biết cả 2 mô hình này sẽ nhớ được các thông tin trong quá khứ và dùng nó để dự đoán hiện tại. Việc của chúng ta là chọn một cấu trúc mô hình phù hợp để tối ưu hóa thời gian cũng như đạt được kết quả tốt nhất.

3.3.1 LSTM

Với mô hình LSTM ta sử dụng cấu trúc sau: input_size= số thuốc tính đã chọn, hidden_layer_size = 512, output_size =1, n_layers = 1. Đây là một cấu hình gần như cơ bản nhất, đơn giản nhất cho một mạng LSTM. Với bài toán dự đoán có nhiều hàm tính loss, ở đây ta dùng L1Loss() (MAE Loss), một chỉ số quan trọng là learning_rate ta chọn 0.001, một thuộc tính khác cũng quan trọng, do tính chất thời tiết thất thường sẽ gây ra các ngoại lệ, ta áp dụng L2 Relularization để giản over-fit với chỉ số weight_decay = 0.01

```
input_size = len(features_)
model = WeatherRNN(input_size).to(device)
loss_function = nn.L1Loss()
optimizer = optim.Adam(model.parameters(), lr=0.001, weight_decay=0.01)
# cost = np.array([])
epochs = 1000
```

Hình 3.8: Cấu hình các super parameter của LSTM.

Tiến hành train với 1000 epoch ta được chỉ số loss như sau:

```
Epoch 900, Loss: 0.5129485726356506
Epoch 910, Loss: 0.5069243311882019
Epoch 920, Loss: 0.5326848030090332
Epoch 930, Loss: 0.5593767166137695
Epoch 940, Loss: 0.5223461985588074
Epoch 950, Loss: 0.5056709051132202
Epoch 960, Loss: 0.522179126739502
Epoch 970, Loss: 0.5767115354537964
Epoch 980, Loss: 0.5349583029747009
Epoch 990, Loss: 0.5307326912879944
```

Hình 3.9: Loss của LSTM khi train.

Với một lượng data khoảng 9500 seq và một cấu trúc mô hình đơn giản nhất của LSTM, sự mất mát này là chấp nhận được.Khi test trên bộ dữ liệu test đã được cuẩn hóa dự trên tập train với khoảng 800 seq ta được error là 0.67 với thang đo MAE loss:

```
for i, feature in enumerate(features_):
    X_test[:, i] = scalers[feature].transform(X_test[:,i].reshape(-1, 1)).reshape(-1)
```

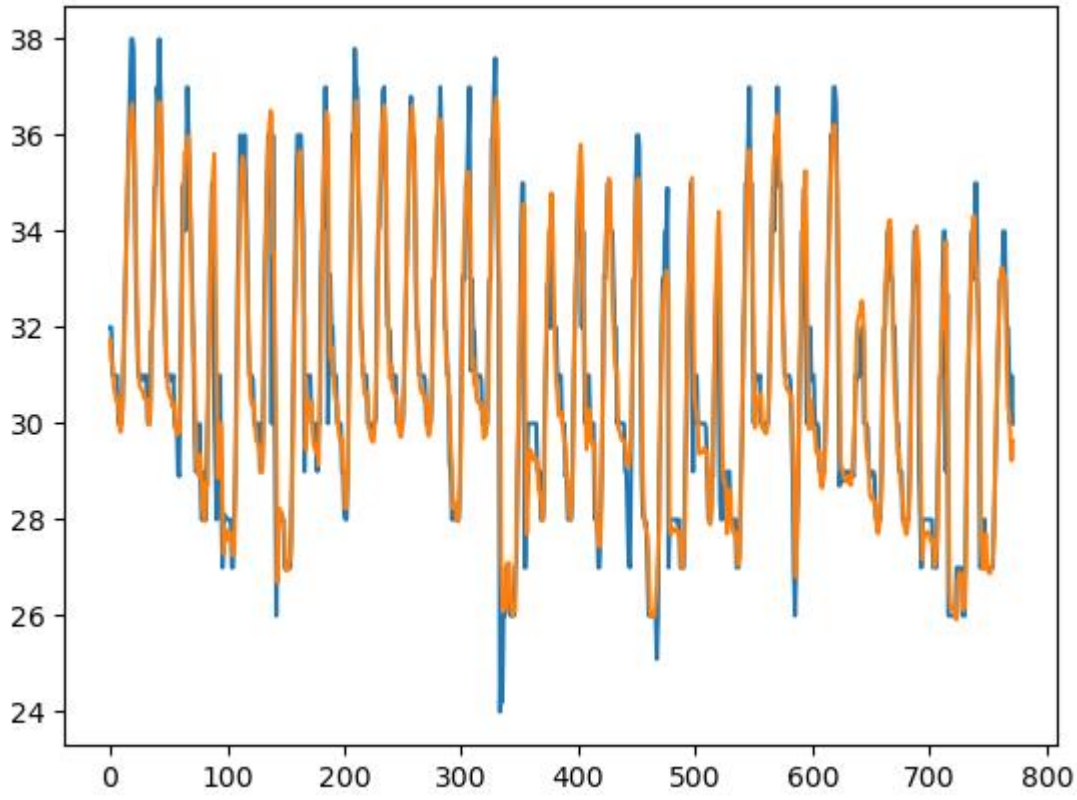
```
X_seq_test, y_seq_test = create_sequences(X_test, y_test, seq_length)
```

```
X_test_t = torch.tensor(X_seq_test, dtype=torch.float32).to(device)
y_test_t = torch.tensor(y_seq_test, dtype=torch.float32).unsqueeze(1).to(device)
```

```
modelx.eval()
with torch.no_grad():
    predictions = modelx(X_test_t)
predictions = predictions.cpu()
print('accuracy: ', mean_absolute_error(y_test_t.cpu(), predictions))
```

```
accuracy:  0.6714883
```

Hình 3.10: Dự đoán cùng với error của LSTM.



Hình 3.11: Đồ thị về giá trị dự đoán (cam) – thực sự (xanh).

Đồ thị biểu diện giữa dự đoán (cam) và thực tế (xanh), chưa hoàn toàn chính xác nhưng đã phần nao bắt được nhịp độ thời tiết.

3.3.2 RNN

Cấu trúc mạng RNNs sẽ được triển khai tương tự với LSTM với hidden_layer_sizee = 512, n_layers=1, output_size =1, cùng sử dụng L1Loss và Adam optimizer với learning_rates = 0.001 và weight_decay = 0.01.

```
input_size = len(features_)
model = WeatherRNN(input_size).to(device)
loss_function = nn.L1Loss()
optimizer = optim.Adam(model.parameters(), lr=0.001, weight_decay=0.01)
```

Hình 3.12: Cấu hình các super parameter của RNN.

Cũng được train trên điều kiện tương tự với 1000 epochs cho ra Loss như sau:

```
torch.Size([9484, 20, 10])
Epoch 0, Loss: 29.138696670532227
Epoch 100, Loss: 6.070045471191406
Epoch 200, Loss: 2.1512768268585205
Epoch 300, Loss: 0.8894159197807312
Epoch 400, Loss: 0.7066817879676819
Epoch 500, Loss: 0.6644251942634583
Epoch 600, Loss: 0.6303276419639587
Epoch 700, Loss: 0.6086217761039734
Epoch 800, Loss: 0.5925964117050171
Epoch 900, Loss: 0.5708107352256775
```

Hình 3.13: Loss của RNN khi train.

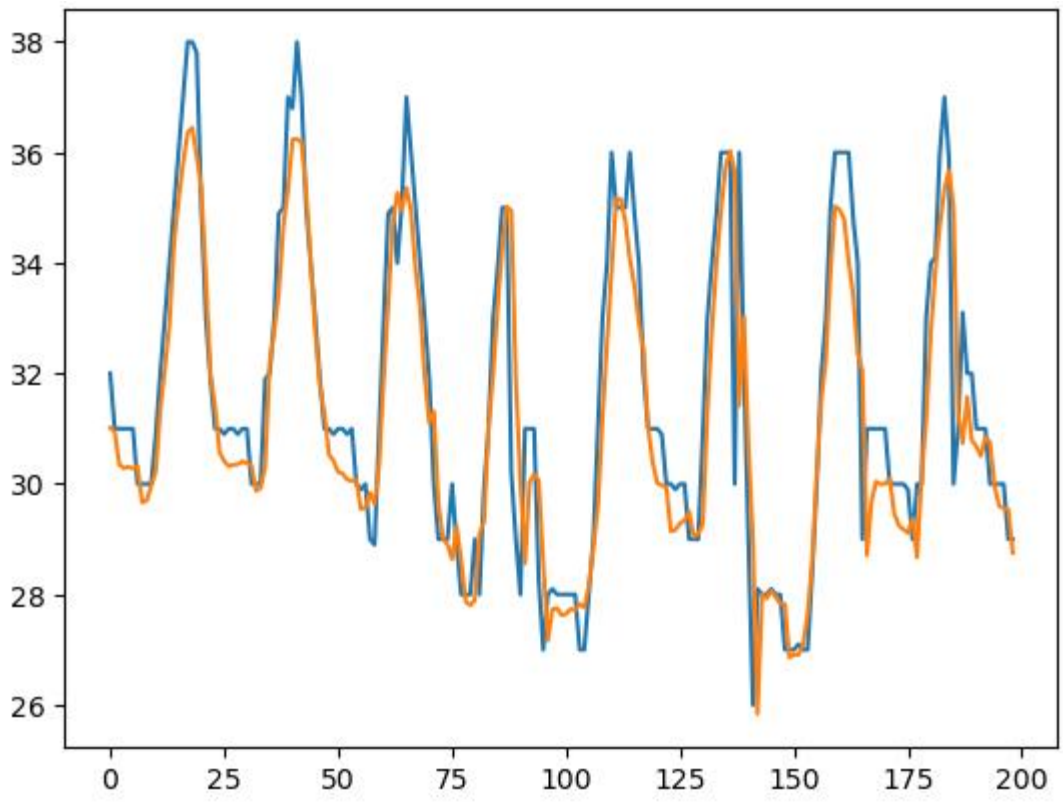
Mạng RNNs cho ra MAE lớn hơn so với LSTM:

```
model.eval()
with torch.no_grad():
    predictions = model(X_test_t)
predictions = predictions.cpu()
print('accuracy: ', mean_absolute_error(y_test_t.cpu(), predictions))
```

```
accuracy:  0.744728
```

Hình 3.14: Dự đoán cùng với error của RNN.

Biểu đồ biểu diễn dự đoán cũng không tốt lắm:



Hình 3.15: Đồ thị về giá trị dự đoán (cam) – thực sự (xanh).

3.4 Kết luận.

Nhìn chung 2 thuật toán cho ra kết quả là tạm chấp nhận được với bộ dữ liệu train khá ít (khoảng 9500 dòng) và cấu trúc mạng được coi là đơn giản nhất. LSTM có độ chính xác cao hơn nhưng thời gian train lâu hơn với cấu trúc có phần phức tạp hơn so với mạng RNNs cơ bản. Sự đánh đổi này là hiển nhiên đối với các mạng học sâu, ta không thể đòi hỏi một mô hình "Ngon – Bổ - Rẻ" được. Với cách tiếp cận này, nếu có nhiều dữ liệu thời tiết hơn từ nhiều năm, mạng có nhiều lớp hơn, nhiều neuron hơn trong mỗi lớp thì chắc chắn sẽ cho ra kết quả chính xác hơn, kéo theo đó là chi phí và thời gian sẽ cao hơn.

Chương 4. Kết luận

4.1 Kết quả đạt được.

Qua quá trình nghiêng cứu, tìm hiểu, môn học đã cơ bản cung cấp được một cái nhìn cụ thể về tầm quan trọng của Data Mining. Được tìm hiểu cặn kẽ về ứng dụng, các thức hoạt động của các thuật toán như phân lớp, gom cụm, …

Từ một bài toán thực tế, sinh viên đã tự thu thập dữ liệu từ các nguồn, biết xử lý dữ liệu, loại bỏ các trường không cần thiết, thực hiện chuẩn hóa bằng MinMaxScaler, StandardScaler, … và chọn ra phương pháp tối ưu nhất.

Biết xây dựng một mô hình mạng neuron để thực hiện dự đoán, thay đổi các siêu tham số để ra được giá trị loss thấp nhất. Lựa chọn tối ưu giữa MAELoss, MSELoss, … Kiểm thử giữa các optimizer như Adam, SGD…

So sánh kết quả của các phương pháp khác nhau (LSTM và RNN) và rút ra các kết luận.

4.2 Hướng phát triển.

Cả 2 mô hình LSTM và RNN vẫn chưa tối ưu được Loss. Trong tương lai sẽ thu thập nhiều dữ liệu hơn, huấn luyện trên một mạng sâu hơn.

Thêm các ứng dụng khác dựa trên data và phương pháp cũ như dự đoán lượng mưa, dự đoán lúc nào sẽ có mưa và dự đoán thời gian mưa kéo dài bao lâu.

Xây dựng bộ giao diện để hiển thị thông tin cho người dùng xem, dùng các API để nhận dữ liệu mới nhất về thời tiết phục vụ cho quá trình dự đoán.

Tài liệu tham khảo

[1] TS. Võ Thị Hồng Thắm, Slide bài giảng Khai thác dữn liệu và Ứng dụng.

[2] Jiawei Han, Micheline Kamber, Jian Pei. “Data mining: Concept and Techniques 3rd Edition”.

[3] MachineLearning cơ bản, https://machinelearningcoban.com/, (accessed 28/5,2024 – 3/6,2024).