**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI: NGHIÊN CỨU MÔ HÌNH DECISION TREE, XÂY DỰNG HỆ THỐNG DỰ BÁO THỜI TIẾT**

Giảng viên hướng dẫn: Th.S TRẦN PHONG NHÃ

Sinh viên thực hiện: NGÔ TÙNG HIỀN

Lớp: CÔNG NGHỆ THÔNG TIN

Khóa:60

Tp. Hồ Chí Minh, năm 2023

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI: NGHIÊN CỨU MÔ HÌNH DECISION TREE, XÂY DỰNG HỆ THỐNG DỰ BÁO THỜI TIẾT**

Giảng viên hướng dẫn: Th.S TRẦN PHONG NHÃ

Sinh viên thực hiện: NGÔ TÙNG HIỀN

Lớp: CÔNG NGHỆ THÔNG TIN

Khóa:60

Tp. Hồ Chí Minh, năm 2023

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH Độc lập – Tự do – Hạnh phúc**



# 

# NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

**BỘ MÔN: CÔNG NGHỆ THÔNG TIN**

**-------**\*\*\***-------**

**Mã sinh viên:** 6051071039 **Họ tên SV:** Ngô Tùng Hiền

**Khóa:** 60 **Lớp:** Công nghệ thông tin

1. **Tên đề tài:**

Nghiên cứu mô hình Decision Tree, xây dựng hệ thống dự báo thời tiết.

1. **Mục tiêu:** Tìm hiểu về ngôn ngữ Python và nghiên cứu một số thuật toán K-Means, Decision Tree về phân tích và dự đoán kết quả. Từ đó ứng dụng vào phân tích và đưa ra các dự đoán về thời tiết dựa trên dataset
2. **Nội dung thực hiện**

* Tìm hiểu về mô hình học máy
* Tìm hiểu ngôn ngữ Python và các thư viện cần sử dụng
* Nghiên cứu bài toán phân tích và dự đoán về nhiệt độ thời tiết
* Áp dụng kiến thức: ứng dụng ngôn ngữ Python và thuật toán vào phân tích và đưa ra dự đoán vềdự báo thời tiết.

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

* Công cụ sử dụng: Google Colab
* Ngôn ngữ: Python

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

* Hiểu và sử dụng được ngôn ngữ lập trình Python
* Hiểu được các thuật toán máy học cần sử dụng
* Cài đặt được môi trường sử dụng ngôn ngữ
* Áp dụng được kiến thức và cho ra kết quả

1. **Kế hoạch thực hiện:** 
   * **Tuần 1- 2 và 3**: Tham khảo và chọn đề tài
   * **Tuần 4**: Đưa ra lựa chọn về đề tài
   * **Tuần 5-6**: Tìm hiểu về ngôn ngữ Python, thư viện sử dụng
   * **Tuần 7 đến 11**: Nghiên cứu các thuật toán máy học và áp dụng kiến thức vào bài toán
   * **Tuần 12:** Chỉnh sửa, viết báo cáo
2. **Giáo viên và cán bộ hướng dẫn**

Họ tên: ThS. TRẦN PHONG NHÃ

Đơn vị công tác: Trường Đại học Giao thông Vận tải Phân hiệu tại TP. Hồ Chí Minh

Điện thoại: Email: [tpnha@utc2.edu.vn](mailto:tpnha@utc2.edu.vn)

|  |  |
| --- | --- |
| **Ngày….. tháng 03 Năm 2023** | **Đã giao nhiệm vụ TKTN** |
| **Trưởng BM Công Nghệ Thông Tin** | **Giảng viên hướng dẫn** |
|  | **Trần Phong Nhã** |

# LỜI CẢM ƠN

Lời đầu tiên, tôi xin cảm ơn tập thể giảng viên **Bộ môn Công nghệ thông tin trường Đại học Giao Thông Vận Tải phân hiệu tại Thành Phố Hồ Chí Minh** đã dạy bảo, truyền tải kiến thức và giúp đỡ em trong suốt quá trình học tập tại trường.

Tôi xin chân thành cảm ơn sự giúp đỡ tận tình của thầy Trần Phong Nhã, người đã định hướng, cung cấp cho tôi nhứng kiến thức, nguồn tài liệu và tận tình hướng dẫn chỉ bảo trong suốt quá trình thực hiện đồ án tốt nghiệp của mình.

Tôi cũng xin gửi lời cảm ơn đến gia đình, bạn bè đã luôn là nguồn động viên to lớn giúp vượt qua những khó khăn trong suốt quá trình học tập.

Mặc dù đã cố gắng hoàn thiện đề tài với tất cả nỗ lực của bản thân, nhưng chắc sẽ không tránh khỏi những thiếu sót. Kính mong thầy cô có thể đóng góp ý kiến để tôi có thể hoàn thiện kiến thức của bản thân.

Cuối cùng, tôi xin chân thành cảm ơn các bạn sinh viên lớp Công Nghệ Thông Tin K60 đã giúp đỡ, chia sẻ và khuyến khích tôi trong quá trình học tập tại trường.

TP.Hồ Chí Minh, ngày..... tháng 06 năm 2023

Sinh viên

**Ngô Tùng Hiền**

# NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

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

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

MỤC LỤC

[NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP iii](#_Toc138196900)

[LỜI CẢM ƠN iii](#_Toc138196901)

[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN iv](#_Toc138196902)

[DANH MỤC CÁC KÍ HIỆU VÀ CHỮ VIẾT TẮT vii](#_Toc138196903)

[MỤC LỤC HÌNH ẢNH viii](#_Toc138196904)

[MỞ ĐẦU 1](#_Toc138196905)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 3](#_Toc138196906)

[**1.1.** **Khái niệm về học máy** 3](#_Toc138196907)

[**1.2.** **Các ứng dụng của máy học** 5](#_Toc138196908)

[1.2.1. Tự động phân loại 5](#_Toc138196909)

[1.2.2. Ứng dụng trong mạng xã hội 5](#_Toc138196910)

[1.2.3. Nhận diện hình ảnh 6](#_Toc138196911)

[1.2.4. Ứng dụng tự động lái xe 6](#_Toc138196912)

[**1.3.** **Các dạng học máy** 6](#_Toc138196913)

[**1.4.** **Các thuật toán phổ biến trong học máy** 9](#_Toc138196914)

[1.4.1. Linear Regression (hồi quy tuyến tính) 9](#_Toc138196915)

[1.4.2. Logictic Regerssion (hồi quy logictic) 9](#_Toc138196916)

[1.4.3. Decision Tree (cây quyết định) 10](#_Toc138196917)

[1.4.4. Support Vector Machine (SVM) 11](#_Toc138196918)

[1.4.5. Naive Bayes 12](#_Toc138196919)

[1.4.6. K-Nearest Neighbors ( KNN ) 15](#_Toc138196920)

[1.4.7. K-Means 15](#_Toc138196921)

[1.4.8. Random Forest 16](#_Toc138196922)

[1.4.9. Dimensionality Reduction 17](#_Toc138196923)

[1.4.10. Thuật toán RNN 18](#_Toc138196924)

[CHƯƠNG 2: PHÂN TÍCH BÀI TOÁN 20](#_Toc138196925)

[**2.1.** **Công cụ hỗ trợ** 20](#_Toc138196926)

[2.1.1. Các khung phần mềm học máy, học sâu 20](#_Toc138196927)

[2.1.2. Một số thư viện sử dụng 22](#_Toc138196928)

[**2.2.** **Các công nghệ hỗ trợ** 25](#_Toc138196929)

[2.2.1. Colaboratory (Google Colab) 25](#_Toc138196930)

[2.2.2. Kaggle 26](#_Toc138196931)

[**2.3.** **Thuật toán sử dụng** 28](#_Toc138196932)

[2.3.1. Thuật toán K-menas Clustering 28](#_Toc138196933)

[2.3.2. Thuật toán Decision Tree 33](#_Toc138196934)

[CHƯƠNG 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ 35](#_Toc138196935)

[**3.1.** **Phương pháp cài đặt thử nghiệm** 35](#_Toc138196936)

[**3.2.** **Giới thiệu bộ dữ liệu** 36](#_Toc138196937)

[**3.3.** **Quá trình phân tích thực nghiệm** 39](#_Toc138196938)

[**3.4.** **Triển khai thực nghiệm** 39](#_Toc138196939)

[3.4.1. Cài đặt thư viện và đọc dữ liệu 40](#_Toc138196940)

[3.4.2. Mô tả thuộc tính 40](#_Toc138196941)

[3.4.3. Trực quan hóa dữ liệu 42](#_Toc138196942)

[3.4.4. Xây dựng mô hình 43](#_Toc138196943)

[KẾT LUẬN 54](#_Toc138196944)

[TÀI LIỆU THAM KHẢO 55](#_Toc138196945)

# DANH MỤC CÁC KÍ HIỆU VÀ CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| STT | Từ viết tắt | Từ đầy đủ |
| 1 | AI | Artificial Intelligence |
| 2 | ML | Machine Learning |
| 3 | SVM | Support Vector Machine |
| 4 | RNN | Recurrent Neural Network |
| 5 | NumPy | Numeric Python |
| 6 | GPU | Graphic Processing Unit |
| 7 | TPU | Tensor Processing Unit |
| 8 | KNN | K-nearest neighbors |
| 9 | API | Application Programming Interface |

# MỤC LỤC HÌNH ẢNH

[Hình 1. 1: *Giới thiệu Machine Learning* 4](#_Toc4961)

[Hình 1. 2: *Các dạng hạt máy* 7](#_Toc30507)

[Hình 1. 3: *Hồi quy tuyến tính* 9](#_Toc13863)

[Hình 1. 4*: Logictic Regerssion* 10](#_Toc8729)

[Hình 1. 5: *Cây quyết định* 11](#_Toc27590)

[Hình 1. 6: *Support Vector Machine* 12](#_Toc3493)

[Hình 1. 7: *K-Nearest Neighbors* 15](#_Toc29957)

[Hình 1. 8: *K-Means* 16](#_Toc15073)

[Hình 1. 9: *Random Forest* 17](#_Toc3207)

[Hình 1. 10: *Dimensionality Reduction* 18](#_Toc10321)

[Hình 1. 11: *Thuật toán chuyển tiếp* 19](#_Toc29551)

[Hình 2. 1: *TensorFlow* 21](#_Toc1172)

[Hình 2. 2: *Các ứng dụng của thư viện NumPy* 23](#_Toc20437)

[Hình 2. 3: *Các biểu đồ trong thư viện Matplotlib* 24](#_Toc12146)

[Hình 2. 4: *Sử dụng google Colab để tạo mô hình* 26](#_Toc21553)

[Hình 3. 1: *Bộ dữ liệu thực nghiệm* 37](#_Toc17441)

[Hình 3. 2: *Sơ đồ phân tích thực nghiệm* 39](#_Toc7922)

[Hình 3. 3: C*ài đặt thư viện* 40](#_Toc32523)

[Hình 3. 4: *Đọc dữ liệu* 40](#_Toc31693)

[Hình 3. 5: *Mô tả thuộc tính* 41](#_Toc8788)

[Hình 3. 6: *Chuyển đổi chuỗi thành đối tượng datetime* 41](#_Toc15500)

[Hình 3. 7: *Nhiệt độ xuyên suốt thời gian* 42](#_Toc22507)

[Hình 3. 8: *Đánh giá V số cụm* 44](#_Toc19821)

[Hình 3. 9: *Cụm nhiệt độ* 44](#_Toc17658)

[Hình 3. 10: *Biểu đồ tần suất nhiệt độ* 46](#_Toc2377)

[Hình 3. 11: *Nhiệt độ trung bình hàng năm* 47](#_Toc25063)

[Hình 3. 12: *Đường xu hướng qua các năm* 48](#_Toc13339)

# 

# MỞ ĐẦU

1. **Lý do chọn đề tài**

Thời tiết là một trong những yếu tố quan trọng ảnh hướng tới đến đời sống của con người. Việc có thể dự báo chính xác thời tiết sẽ giúp cho con người có thể chuẩn bị tốt hơn các hoạt động hàng ngày, đồng thời giảm thiểu các thiệt họa thiên nhiên. Vì vậy, việc xây dựng hệ thống dự báo thời tiết là nhu cầu cấp thiết trong xã hội hiện nay.

Trong đề tài này, sẽ sử dụng ngôn ngữ lập trình Python cùng các thư viện phổ biến như Sciklit-learn, Pandas, NumPY để xây dựng mô hình học máy có khả năng dự báo thời tiết chính xác. Đầu tiên, sẽ thu thập và tiền xử lý dữ liệu thời tiết từ các nguồn khác như trang web dự báo thời tiết, các đài khí tượng,.. Sau đó, sẽ thực hiện huấn luyện mô hình học máy trên dữ liệu đã thu thập. Một số mô hình học máy phổ biến như Decision Tree, Random Forest, SVM,… sẽ được sủ dụng so sánh và đánh giá độ chính xác của các mô hình. Cuối cùng, sẽ triển khai mô hình trên ứng dụng web để người dùng có thể tra cứu và nhận được dự báo thời tiết chính xác và nhanh chóng.

Việc xây dựng hệ thống dự báo thời tiết sử dụng học máy với ngôn ngữ Python không chỉ giúp cho có thể áp dụng các kỹ thuật mới nhất trong lĩnh vực khoa học dữ liệu và trí tuệ nhân tạo mà còn giúp cho con người có thể chuẩn bị tốt hơn cho các hoạt động hàng ngày và giảm thiệt hạ do thiên tai.

1. **Mục tiêu và nhiệm vụ đồ án**

Tìm hiểu về ngôn ngữ Python và nghiên cứu thuật toán máy học về phân tích và dự đoán kết quả như Decision Tree, K-means. Tù đó ứng dụng vào phân tích và đưa ra các dự đoán về thời tiết dựa trên dataset được lấy từ trang Kaggle.com

1. **Phương pháp nghiên cứu**

* Phương pháp thu thập dữ liệu
* Phương pháp phân tích và tổng hợp
* Phương pháp toán học
* Phương pháp nghiên cứu định tính
* Phương pháp thực nghiệm.

1. **Bố cục đồ án**

Bố cục đồ án được chia làm 4 phần và bao gồmm những nội dung như sau:

* Tổng quan – Giới thiệu tổng quan về đề tài đồ án tốt nghiệp
* Chương 1: Cơ sở lý thuyết: Tìm hiểu kỹ thuật mô hình học máy Decision Tree và K-menas. Các khái niệm, thuật toán liên quan đế đề tài nghiên cứu.
* Chương 2: Phân tích bài toán: Phân tích dữ liệu, đưa ra các công cụ hỗ trợ và phương pháp đánh giá mô hình
* Chương 3: Thực nghiệm và đánh giá kết quả: Xây dựng cài đặt mô hình, thực hiện thử nghiệm dự đoán.
* Kết luận: Tổng kết lại quá trình nghiên cứu và thực nghiệm, những kết quả đạt được, hướng phát triển.
* Tài liệu tham khảo

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

* 1. **Khái niệm về học máy**

Học máy ( Machine Learning) là một nhánh nhỏ của ngành trí tuệ nhân tạo (AI) và khoa học máy tính. Nó tập trung vào việc xử lý data và thuật toán để bắt chước cách mà con người học. Từ đó cải thiện độ chính xác của các dự đoán mà nó đưa ra.

Học máy là một thành phần vô cùng quan trọng trong sự phát triển của khoa học dữ liệu. Thông qua việc sử dụng các phương pháp thống kê và thuật toán, chúng được huấn luyện để đưa ra các phân loại, dự đoán và khám phá những tri thức mới trong các dự án khai phá dữ liệu. Những tri thức này sẽ tác động đến những quyết định trong các ứng dụng và các doanh nghiệp.

Khi internet ngày càng phổ biến, mọi người trong số vẫn thường xuyên lên mạng để cập nhập tin tức, giải trí, mua sắm,... Chính những hoạt động này đã sản sinh ra một lượng lớn dữ liệu. Khi dữ liệu lớn tiếp tục mở rộng và phát triển, nhu cầu của thị trường với các nhà khoa học dữ liệu ngày càng tăng. Nó cũng đòi hỏi nhà khoa học phải hỗ trợ xác định các câu hỏi của doanh nghiệp và đưa ra quyết định phù hợp nhất.

Học máy là một nhánh nghiên cứu rất quan trọng của trí tuệ nhân tạo với khá nhiều ứng dụng thành công trong thực tế. Hiện nay, học máy là một trong những lĩnh vực phát triển mạnh nhất của trí tuệ nhân tạo. Có một số lý do giải thích cho sự cần thiết và phát triển của học máy:

**Thứ nhất,** rất khó xây dựng hệ thống thông minh có thể thực hiện các công việc liên quan đến trí tuệ như thị giác máy, xử lý ngôn ngữ tự nhiên mà không sử dụng tới kinh nghiệm và quá trình học. Thông thường, khi viết chương trình, cần có thuật toán rõ ràng để chuyển đổi đầu vào thành đầu ra. Tuy nhiên, trong nhiều bài toán, rất khó để xây dựng được thuật toán như vậy. Như trong ví dụ về nhận dạng chữ ở trên, người bình thường có khả năng nhận dạng các chữ rất tốt nhưng rất khó để giải thích vì sao từ đầu vào là ảnh lại kết luận được đây là ký tự cụ thể nào. Học máy cho phép tìm ra giải pháp cho những trường hợp như vậy dựa trên dữ liệu, chẳng hạn bằng cách tìm ra điểm chung và riêng giữa rất nhiều ảnh chụp các ký tự.

**Thứ hai,** nhiều ứng dụng đòi hỏi chương trình máy tính phải có khả năng thích nghi. Ví dụ, hành vi mua sắm của khách hàng có thể thay đổi theo thời điểm cụ thể trong ngày, trong năm, hoặc theo tuổi tác. Việc xây dựng thuật toán cố định cho những ứng dụng cần thích nghi và thay đổi là không phù hợp. Học máy mang lại khả năng thích nghi nhờ phân tích dữ liệu thu thập được.

**Thứ ba,** việc tìm được chuyên gia và thu thập được tri thức cần thiết cho việc thiết kế thuật toán để giải quyết các vấn đề tương đối khó, trong khi dữ liệu ngày càng nhiều và có thể thu thập dễ dàng hơn. Khả năng lưu trữ và tính toán của máy tính cũng ngày càng tăng, cho phép thực hiện thuật toán học máy trên dữ liệu có kích thước lớn.

**Cuối cùng,** bản thân khả năng học là một hoạt động trí tuệ quan trọng của con người, do vậy học tự động hay học máy luôn thu hút được sự quan tâm khi xây dựng hệ thống thông minh.



Hình 1. 1: Giới thiệu Machine Learning

* 1. **Các ứng dụng của máy học**
     1. Tự động phân loại

Phân loại tin tức là một ứng dụng điểm chuẩn khác của phương pháp học máy. Như một vấn đề thực tế là bây giờ khối lượng thông tin đã tăng lên rất nhiều trên web. Tuy nhiên, mỗi người có sở thích hoặc lựa chọn cá nhân của mình. Vì vậy, để chọn hoặc thu thập một phần thông tin phù hợp trở thành một thách thức đối với người dùng từ vô số nội dung trên trang web.

Phân loại các danh mục một cách rõ ràng, dễ điều hướng giúp cho các khách hàng mục tiêu chắc chắn sẽ tăng khả năng truy cập các trang tin tức. Hơn nữa, độc giả hoặc người dùng có thể tìm kiếm tin tức cụ thể một cách hiệu quả và nhanh chóng.

Có một số phương pháp học máy trong mục đích này, tức là, máy vectơ hỗ trợ, naive Bayes, k-nearest neighbor, v.v.

* + 1. Ứng dụng trong mạng xã hội

Học máy đang được sử dụng trong một loạt các ứng dụng ngày nay. Một trong những ví dụ nổi tiếng nhất là Facebook News Feed. Nguồn cấp tin tức sử dụng học máy để cá nhân hóa từng nguồn cấp dữ liệu thành viên.

Nếu một thành viên thường xuyên dừng lại để đọc hoặc thích một bài đăng của một người bạn cụ thể, News Feed sẽ bắt đầu hiển thị nhiều hơn về hoạt động của người bạn đó trước đó trong nguồn cấp dữ liệu.

Đằng sau hệ thống ấy, phần mềm sử dụng phân tích thống kê và phân tích dự đoán để xác định các mẫu trong dữ liệu người dùng và sử dụng các mẫu đó để điền vào News Feed. Nếu thành viên không còn dừng lại để đọc, thích hoặc bình luận trên các bài đăng của bạn bè, dữ liệu mới đó sẽ được bao gồm trong tập dữ liệu và News Feed sẽ điều chỉnh tương ứng.

Không chỉ riêng facebook, ta có thể bắt gặp những tính năng tương tự đó qua các mạng xã hội khác như google, instagram,...

* + 1. Nhận diện hình ảnh

Nhận dạng hình ảnh là một trong những ví dụ về máy học và trí tuệ nhân tạo phổ biến nhất. Về cơ bản, nó là một cách tiếp cận để xác định và phát hiện các đặc trưng của một đối tượng trong hình ảnh kỹ thuật số. Hơn nữa, kỹ thuật này có thể được sử dụng để phân tích sâu hơn, chẳng hạn như nhận dạng mẫu, nhận diện hình khuôn, nhận dạng khuôn mặt, nhận dạng ký tự quang học và nhiều hơn nữa,..

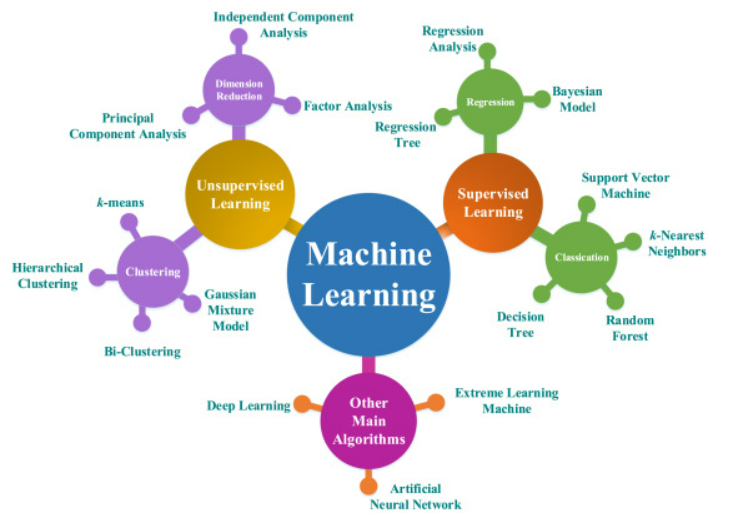
* + 1. Ứng dụng tự động lái xe

Một trong những ứng dụng thú vị nhất của Machine learning là ô tô tự lái. Machine learning đóng một vai trò quan trọng trong việc vận hành ô tô tự lái. Tesla, công ty sản xuất xe hơi nổi tiếng đang tiến hành các nghiên cứu về xe tự lái bằng phương pháp học không giám sát (unsupervised learning method )để đào tạo các mô hình ô tô tự động phát hiện ra người và đồ vật khi đang lái xe.

* 1. **Các dạng học máy**

Khi thiết kế và xây dựng hệ thống học máy cần quan tâm tới những yếu tố sau.

* Thứ nhất, kinh nghiệm hoặc dữ liệu cho học máy được cho dưới dạng nào?
* Thứ hai, lựa chọn biểu diễn cho hàm đích ra sao? Hàm đích có thể biểu diễn dưới dạng hàm đại số thông thường nhưng cũng có thể biểu diễn dưới những dạng khác như dạng cây, dạng mạng nơ ron, công thức xác suất .v.v



Hình 1. 2: Các dạng hạt máy

* + 1. Học có giám sát

Là dạng học máy trong đó cho trước tập dữ liệu huấn luyện dưới dạng các ví dụ cùng với giá trị đầu ra hay giá trị đích. Dựa trên dữ liệu huấn luyện, thuật toán học cần xây dựng mô hình hay hàm đích để dự đoán giá trị đầu ra (giá trị đích) cho các trường hợp mới.

* Nếu giá trị đầu ra là rời rạc thì học có giám sát được gọi là phân loại hay phân lớp (classification).
* Nếu đầu ra nhận giá trị liên tục, tức đầu ra là số thực, thì học có giám sát được gọi là hồi quy (regression). Trong phần tiếp theo, ta sẽ xem xét chi tiết hơn về học có giám sát.
  + 1. Học không giám sát (un-supervised leanrning)

Là dạng học máy trong đó các ví dụ được cung cấp nhưng không có giá trị đầu ra hay giá trị đích.

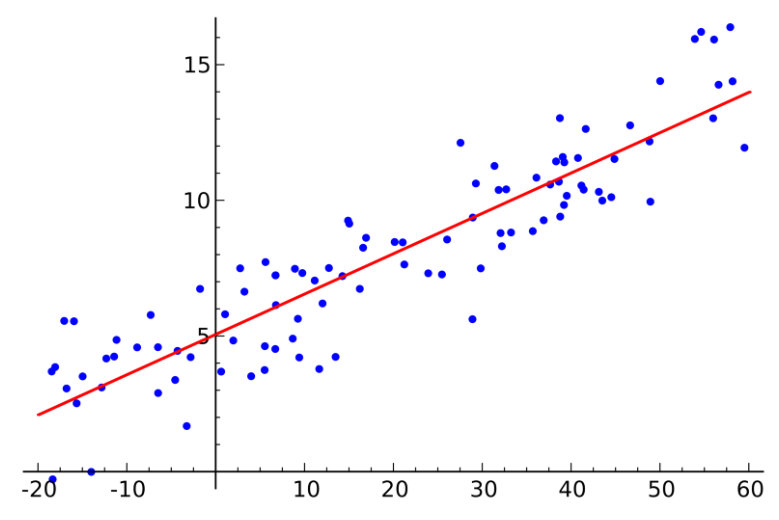
* Thay vì xác định giá trị đích, thuật toán học máy dựa trên độ tương tự giữa các ví dụ để xếp chúng thành những nhóm, mỗi nhóm gồm các ví dụ tương tự nhau. Hình thức học không giám sát như vậy gọi là phân cụm (clustering). Ví dụ, chỉ bằng cách quan sát hoặc đo chiều cao của mọi người, dần dần ta học được khái niệm “người cao” và “người thấp”, và có thể xếp mọi người vào hai cụm tương ứng.
* Ngoài phân cụm, một dạng học không giám sát phổ biến khác là phát hiện luật kết hợp (association rule). Luật kết hợp có dạng P(A | B), cho thấy xác suất hai tính chất A và B xuất hiện cùng với Ví dụ, qua phân tích dữ liệu mua hàng ở siêu thị, ta có luật P(Bơ | Bánh mỳ) =80%, có nghĩa là 80% những người mua bánh mỳ cũng mua bơ.
  + 1. Học tăng cường (reinforcement learning)

Đối với dạng học này, kinh nghiệm không được cho trực tiếp dưới dạng đầu vào/đầu ra cho mỗi trạng thái hoặc mỗi hành động. Thay vào đó, hệ thống nhận được một giá trị khuyến khích (reward) là kết quả cho một chuỗi hành động nào đó. Thuật toán cần học cách hành động để cực đại hóa giá trị khuyển khích. Ví dụ của học khuyến khích là học đánh cờ, trong đó hệ thống không được chỉ dẫn nước đi nào là hợp lý cho từng tình huống mà chỉ biết kết quả toàn ván cờ. Như vậy, các chỉ dẫn về nước đi được cho một cách gián tiến và có độ trễ dưới dạng giá trị thưởng. Nước đi tốt là nước đi nằm trong một chuỗi các nước đi dẫn tới kết quả thắng toàn bộ ván cờ.

Trong các dạng học máy, học có giám sát là dạng phổ biến, có nhiều thuật toán liên quan và nhiều ứng dụng nhất

* 1. **Các thuật toán phổ biến trong học máy**
     1. Linear Regression (hồi quy tuyến tính)

Nếu bạn biết thống kê, bạn có thể đã nghe nói về hồi quy tuyến tính trước đây. *Bình phương nhỏ nhất* là một phương pháp để thực hiện hồi quy tuyến tính. Bạn có thể suy nghĩ về hồi quy tuyến tính như là nhiệm vụ kẻ một đường thẳng đi qua một tập các điểm. Có rất nhiều chiến lược có thể thực hiện được, và chiến lược “bình phương nhỏ nhất” sẽ như thế này – Bạn có thể vẽ một đường thẳng, và sau đó với mỗi điểm dữ liệu, đo khoảng cách thẳng đứng giữa điểm và đường thẳng. Đường phù hợp nhất sẽ là đường mà các khoảng cách này càng nhỏ càng tốt.

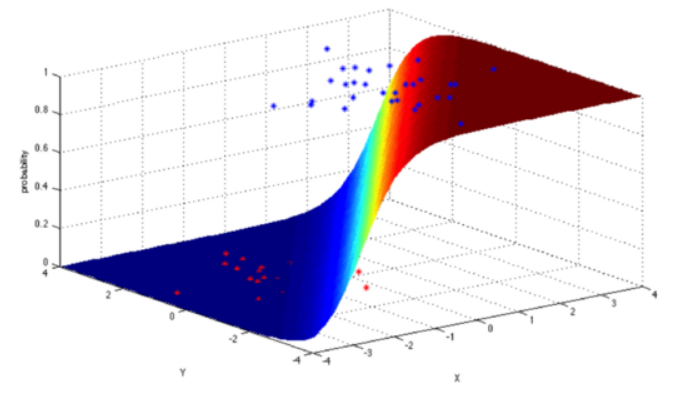


Hình 1. 3: Hồi quy tuyến tính

Một số ví dụ là người ta có thể sử dụng mô hình này để dự đoán giá cả (nhà đất, chứng khoán), điểm số,…

* + 1. Logictic Regerssion (hồi quy logictic)

Hồi quy logistic là một cách thống kê mạnh mẽ để mô hình hóa một kết quả nhị thức với một hoặc nhiều biến giải thích. Nó đo lường mối quan hệ giữa biến phụ thuộc phân loại và một hoặc nhiều biến độc lập bằng cách ước tính xác suất sử dụng một hàm logistic, là sự phân bố tích lũy logistic.

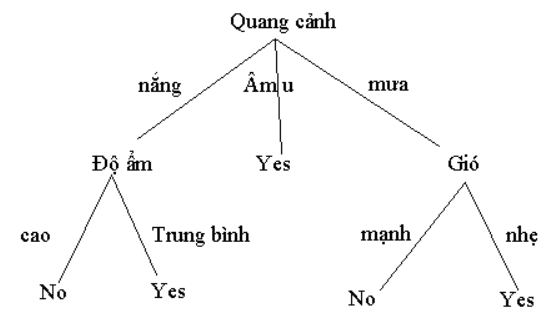


Hình 1. 4: Logictic Regerssion

Thuật toán này được sử dụng trong một số trường hợp:

* Điểm tín dụng ( quyết định có cho khách hàng vay vốn hay không)
* Đo mức độ thành công của chiến dịch marketing
* Dự đoán doanh thu của một sản phẩm nhất định
* Dự đoán động đất …
  + 1. Decision Tree (cây quyết định)

Cây quyết định là công cụ hỗ trợ quyết định sử dụng biểu đồ dạng cây hoặc mô hình của các quyết định và kết quả có thể xảy ra của chúng, bao gồm kết quả sự kiện ngẫu nhiên, chi phí tài nguyên và lợi ích. Dưới đây là một ví dụ điển hình của cây quyết định:



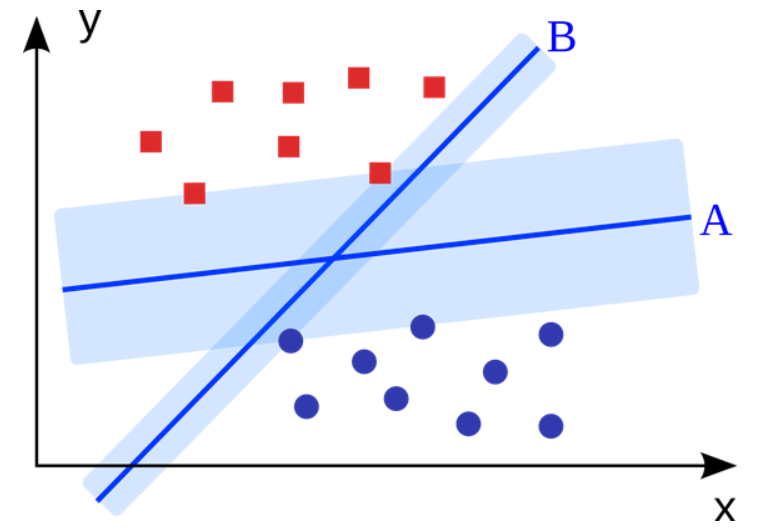
Hình 1. 5: Cây quyết định

Cây quyết định này cho ta gợi ý về việc có đi đá bóng hay không. Ví dụ, quang cảnh có nắng, độ ẩm trung bình thì tôi sẽ đi đá bóng. Ngược lại, nếu trời mưa, gió mạnh thì tôi sẽ không đi đá bóng nữa.

Cây quyết định tuy là mô hình khá cũ, khá đơn giản những vẫn còn được ứng dụng khá nhiều và hiệu quả. Đứng dưới góc nhìn thực tế, cây quyết định là một danh sách tối thiểu các câu hỏi dạng yes/no mà người ta phải hỏi, để đánh giá xác suất đưa ra quyết định đúng đắn.

* + 1. Support Vector Machine (SVM)

SVM là phương pháp phân loại nhị phân. Cho một tập các điểm thuộc 2 loại trong môi trường N chiều, SVM cố gắng tìm ra N-1 mặt phẳng để phân tách các điểm đó thành 2 nhóm. Ví dụ, cho một tập các điểm thuộc 2 loại như hình bên dưới, SVM sẽ tìm ra một đường thẳng nhằm phân cách các điểm đó thành 2 nhóm sao cho khoảng cách giữa đường thẳng và các điểm xa nhất có thể



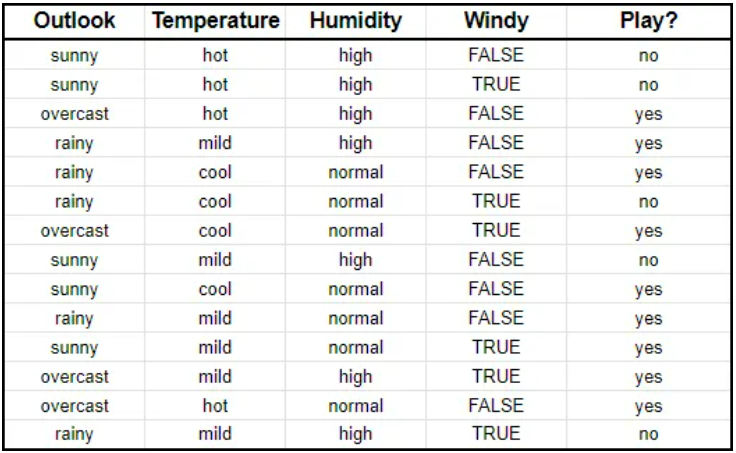
Hình 1. 6: Support Vector Machine

Xét về quy mô, một số vấn đề lớn nhất đã được giải quyết bằng cách sử dụng SVM (với việc thực hiện sửa đổi phù hợp) ví dụ như hiển thị quảng cáo, phát hiện giới tính dựa trên hình ảnh, phân loại hình ảnh có quy mô lớn …

* + 1. Naive Bayes

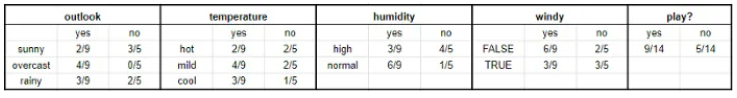
Naive Bayes là một thuật toán phân loại. Điều này có nghĩa là Naive Bayes được sử dụng khi biến đầu ra là rời rạc.

Naive Bayes có vẻ là một thuật toán khó vì nó yêu cầu kiến thức toán học sơ bộ về xác suất có điều kiện và Định lý Bayes, nhưng đó là một khái niệm cực kỳ đơn giản và “ngây thơ”. Hãy xem giải thích về thuật toán này bằng một ví dụ:



Giả sử có dữ liệu đầu vào về các đặc điểm của thời tiết (triển vọng – outlook, nhiệt độ – temperature, độ ẩm – Humidity, gió – Windy) và liệu bạn có chơi gôn hay không (tức là cột cuối cùng).

Điều mà Naive Bayes làm về cơ bản là so sánh tỷ lệ giữa mỗi biến đầu vào và các danh mục trong biến đầu ra. Điều này có thể được hiển thị trong bảng dưới đây.



Lấy một ví dụ trong bảng trên để bạn hiểu, trong phần nhiệt độ, trời nóng trong hai ngày trong số chín ngày bạn chơi gôn (tức là có).

Theo thuật ngữ toán học, bạn có thể viết giá trị này là xác suất trời nóng khi bạn chơi gôn . Kí hiệu toán học là P (hot | yes). Đây được gọi là xác suất có điều kiện và là điều cần thiết để hiểu phần còn lại của những gì sắp xem sau đây.

Sau khi có được điều này, bạn có thể dự đoán liệu mình có chơi gôn hay không với bất kỳ sự kết hợp nào của các đặc điểm thời tiết.

Hãy tưởng tượng rằng có một ngày mới với những đặc điểm sau:

* Triển vọng: nắng
* Nhiệt độ: ôn hòa
* Độ ẩm: bình thường
* Windy: không (false)

**Đầu tiên**, chúng tôi sẽ tính xác suất bạn sẽ chơi gôn với X, P (yes | X), tiếp theo là xác suất bạn sẽ không chơi gôn với X, P (no | X).

Sử dụng biểu đồ trên, có thể nhận được thông tin sau:

**P( yes) = 9/14**

**P(outlook = sunny|yes) =2/9**

**P(temperature = mild|yes) = 4/9**

**P(humidity = normal|yes) = 6/9**

**P(windy = false|yes) = 6/9**

Bây giờ có thể chỉ cần nhập thông tin này vào công thức sau:

**P(yes|X) α P(X|y) \* P(y)**

**P(yes|X) α P(x1|y) \* P(x2|y) \* P(x3|y) \* P(x4|y) \* P(y)**

**P(yes|X) α P(sunny|yes) \* P(mild|yes) \* P(normal|yes) \* P(false|yes) \* P(yes)**

**P(yes|X) α 2/9 \*4/9 \* 6/9 \* 9/14**

**P(yes|X) α 0.0282**

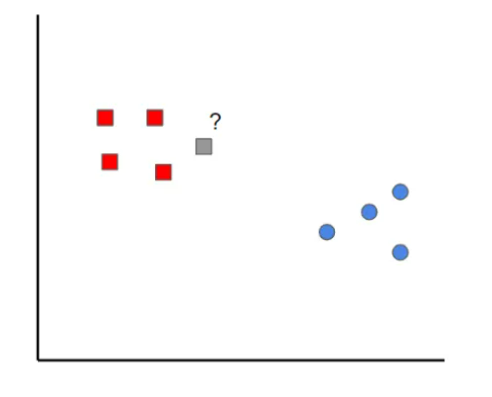
Tương tự, bạn sẽ hoàn thành trình tự các bước tương tự cho P (no | X)

**P( no|X) α 0.0069**

Vì P (yes | X)> P (no | X), nên bạn có thể dự đoán rằng người này sẽ chơi gôn với điều kiện là trời nắng, nhiệt độ ôn hòa, độ ẩm bình thường và không có gió.

Đây là bản chất của Naive Bayes

* + 1. K-Nearest Neighbors ( KNN )



Hình 1. 7: K-Nearest Neighbors

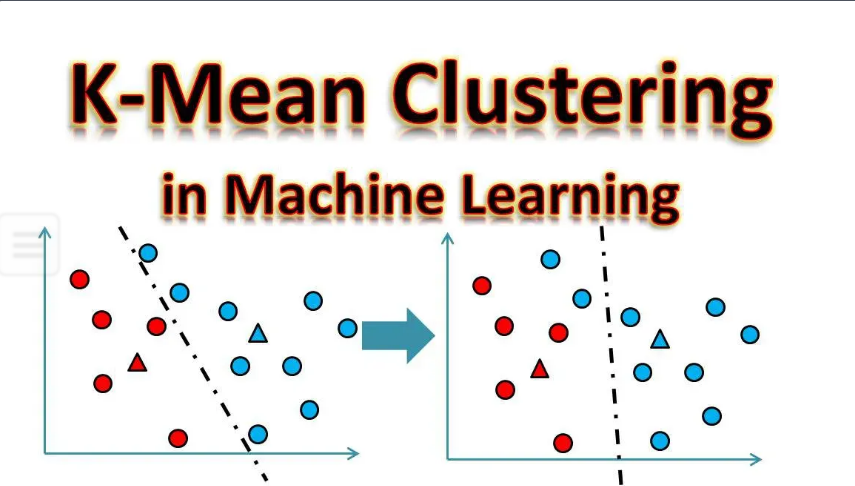
K-nearest neighbors là một ý tưởng đơn giản. Đầu tiên, bạn bắt đầu với dữ liệu đã được phân loại (tức là các điểm dữ liệu màu đỏ và xanh lam). Sau đó, khi bạn thêm một điểm dữ liệu mới, bạn phân loại nó bằng cách xem các điểm được phân loại gần k nhất. Lớp nào nhận được nhiều phiếu bầu nhất sẽ quyết định điểm mới được xếp vào loại nào.

Trong trường hợp này, nếu đặt k = 1, có thể thấy rằng điểm gần nhất đầu tiên với mẫu màu xám là một điểm dữ liệu màu đỏ. Do đó, điểm sẽ được phân loại là màu đỏ.

Một điều cần lưu ý là nếu giá trị của k được đặt quá thấp, nó có thể bị ngoại lệ. Mặt khác, nếu giá trị của k được đặt quá cao thì nó có thể bỏ qua các lớp chỉ có một vài mẫu

* + 1. K-Means

K-Means là một thuật toán học tập không giám sát để giải quyết các vấn đề phân cụm. Các tập dữ liệu được phân loại thành một số cụm cụ thể (hãy gọi số đó là K) theo cách mà tất cả các điểm dữ liệu trong một cụm là đồng nhất và không đồng nhất với dữ liệu trong các cụm khác.

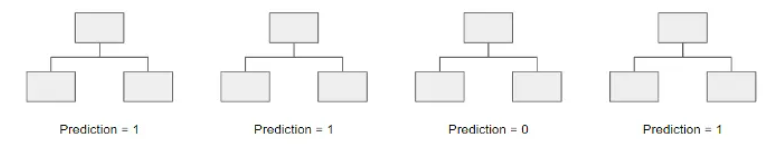


Hình 1. 8: K-Means

Cách K-mean tạo thành các cụm:

* Thuật toán K-mean chọn K số điểm cho mỗi cụm, gọi là centroid.
* Mỗi điểm dữ liệu tạo thành một cụm với các trung tâm gần nhất, tức là cụm K.
* Tạo ra các trung tâm mới dựa trên các cụm thành viên hiện có.
* Với những trung tâm mới này, khoảng cách gần nhất cho mỗi điểm dữ liệu được xác định. Quá trình này được lặp lại cho đến khi các trung tâm không thay đổi
  + 1. Random Forest

Random Forest là một kỹ thuật học tập tổng hợp được xây dựng dựa trên các cây quyết định. Random Forest liên quan đến việc tạo nhiều cây quyết định bằng cách sử dụng tập dữ liệu khởi động của dữ liệu gốc và chọn ngẫu nhiên một tập hợp con các biến ở mỗi bước của cây quyết định. Sau đó, mô hình sẽ chọn chế độ (mode) của tất cả các dự đoán của mỗi cây quyết định (bagging). Bằng cách dựa trên mô hình “đa số thắng”, nó làm giảm nguy cơ mắc lỗi từ một cây riêng lẻ



Hình 1. 9: Random Forest

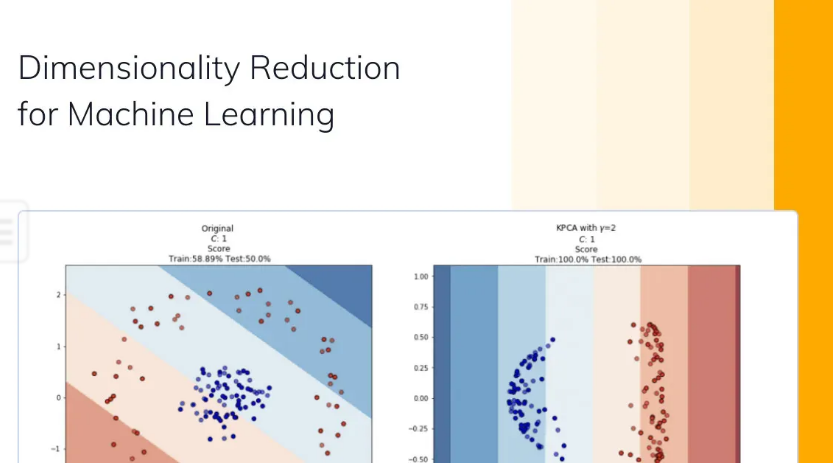
Ví dụ, nếu tạo ra một cây quyết định, cây thứ ba, nó sẽ dự đoán bằng 0. Nhưng nếu dựa vào chế độ (mode) của cả 4 cây quyết định, giá trị dự đoán sẽ là 1. Đây là sức mạnh của các Random Forest

* + 1. Dimensionality Reduction

Trong thế giới ngày nay, một lượng lớn dữ liệu đang được các công ty, cơ quan chính phủ và tổ chức nghiên cứu lưu trữ và phân tích. Các dữ liệu thô này chứa rất nhiều thông tin, thách thức ở đây là phải xác định được các mẫu và biến quan trọng.

Nói một cách đơn giản, Dimensionality Reduction là việc chuyển đổi dữ liệu từ không gian chiều cao thành không gian chiều thấp để biểu diễn chiều thấp giữ lại một số thuộc tính có ý nghĩa của dữ liệu ban đầu.

Các thuật toán giảm kích thước, hay giảm thứ nguyên như Decision Tree, Factor Analysis, Missing Value Ratio và Random Forest có thể giúp bạn tìm thấy các chi tiết có liên quan.



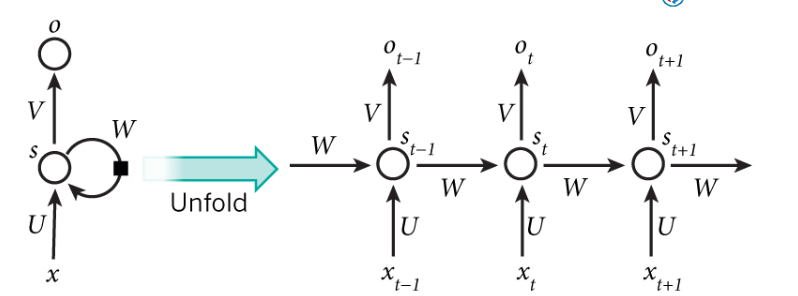
Hình 1. 10: Dimensionality Reduction

* + 1. Thuật toán RNN

Recurrent Neural Network (RNN) là một loại mạng nơ-ron nhân tạo chủ yếu được sử dụng trong nhận dạng giọng nói và xử lý ngôn ngữ tự nhiên (NLP). RNN được sử dụng trong học tập sâu và trong việc phát triển các mô hình bắt chước hoạt động của các tế bào thần kinh trong não người.

Mạng lặp lại được thiết kế để nhận dạng các mẫu trong chuỗi dữ liệu, chẳng hạn như văn bản, bộ gen, chữ viết tay, lời nói và dữ liệu chuỗi thời gian số phát ra từ cảm biến, thị trường chứng khoán và các cơ quan chính phủ.

Ý tưởng chính của RNN (Recurrent Neural Network) là sử dụng chuỗi các thông tin. Trong các mạng nơ-ron truyền thống tất cả các đầu vào và cả đầu ra là độc lập với nhau. Tức là chúng không liên kết thành chuỗi với nhau. Nhưng các mô hình này không phù hợp trong rất nhiều bài toán. Ví dụ, nếu muốn đoán từ tiếp theo có thể xuất hiện trong một câu thì ta cũng cần biết các từ trước đó xuất hiện lần lượt thế nào chứ nhỉ? RNN được gọi là hồi quy (Recurrent) bởi lẽ chúng thực hiện cùng một tác vụ cho tất cả các phần tử của một chuỗi với đầu ra phụ thuộc vào cả các phép tính trước đó. Nói cách khác, RNN có khả năng nhớ các thông tin được tính toán trước đó. Trên lý thuyết, RNN có thể sử dụng được thông tin của một văn bản rất dài, tuy nhiên thực tế thì nó chỉ có thể nhớ được một vài bước trước đó (ta cùng bàn cụ thể vấn đề này sau) mà thôi. Về cơ bản một mạng RNN có dạng như sau:



Hình 1. 11: Thuật toán chuyển tiếp

Mô hình trên mô tả phép triển khai nội dung của một RNN. Triển khai ở đây có thể hiểu đơn giản là ta vẽ ra một mạng nơ-ron chuỗi tuần tự

# CHƯƠNG 2: PHÂN TÍCH BÀI TOÁN

* 1. **Công cụ hỗ trợ**
     1. Các khung phần mềm học máy, học sâu

Bản chất của bài toán Deep learning: Bạn có dữ liệu, bạn muốn máy tính học được các mô hình (model) từ dữ liệu, sau đó dùng mô hình đấy để dự đoán được các dữ liệu mới. Các bước cơ bản làm một bài toán deep learning:

* Xây dựng bài toán
* Chuẩn bị dữ liệu (dataset)
* Xây dựng model
* Định nghĩa loss function
* Thực hiện backpropagation và áp dụng gradient descent để tìm các parameter gồm weight và bias để tối ưu loss function.
* Dự đoán dữ liệu mới bằng model với các hệ số tìm được ở trên

Các framework về deep learning ra đời với các đặc điểm giúp hỗ trợ người dùng:

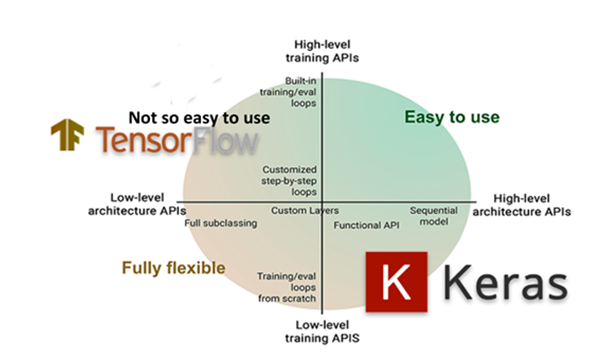
* Người dùng chỉ cần định nghĩa model và loss function, framework sẽ lo phần backpropagation.
* Việc định nghĩa layer, activation function, loss function đơn giản hơn cho người dùng. Ví dụ để thêm layer trong neural network chỉ cần báo là layer có bao nhiêu node và dùng hàm activation gì.
* Tối ưu việc tính toán trên CPU và GPU.

Hàng loạt các thư viện hỗ trợ deep learning ra đời cùng với sự phát triển mạnh mẽ của phần cứng trong việc thiết kế GPU, các mô hình mạng nơ-ron ngày càng được xây dựng lớn hơn và sâu hơn để có thể đảm bảo được yêu cầu về độ chính xác cho các bài toán nhất định. Vì thế, việc tính toán các tham số cho mạng ngày càng phức tạp và rất khó để có thể xử lý được việc huấn luyện một mô hình xây dựng thủ công. Do vậy, các khung phần mềm hỗ trợ lập trình các mạng học sâu ra đời, giúp ta trừu tượng hóa các bước xử lý toán ẩn bên dưới và cung cấp các giao diện lập trình đơn giản, dễ sử dụng trong quá trình đào tạo. Trong đề tài này, khung phần mềm ta sử dụng là Karas.

Keras là một giao diện lập trình ứng dụng (API) mạng thần kinh cho Python được tích hợp chặt chẽ với TensorFlow, được sử dụng để xây dựng các mô hình học máy. Các mô hình của Keras cung cấp một cách đơn giản, thân thiện với người dùng để xác định một mạng nơ-ron.

TensorFlow là một bộ thư viện mã nguồn mở để tạo và làm việc với mạng nơ-ron, chẳng hạn như những thư viện được sử dụng trong các dự án Học máy (ML) và Học sâu.

Mặt khác, Keras là một API cấp cao chạy trên TensorFlow. Keras đơn giản hóa việc triển khai các mạng nơ-ron phức tạp với khuôn khổ dễ sử dụng của nó.



Hình 2. 1: TensorFlow

Keras được coi là một thư viện ‘high-level’ với phần ‘low-level’ (còn được gọi là *backend*) có thể là TensorFlow, CNTK, hoặc Theano. Keras ưu tiên trải nghiệm của người lập trình:

* Keras đã được sử dụng rộng rãi trong doanh nghiệp và cộng đồng nghiên cứu
* Keras tập trung vào trải nghiệm người dùng
* Chạy liền mạch trên CPU và GPU
* Keras hỗ trợ đa backend engines và không giới hạn bạn vào một hệ sinh thái đơn giản hơn TensorFlow rất nhiều.
  + 1. Một số thư viện sử dụng
       1. Thư viện NumPy ( Numeric Python)

Numpy (Numeric Python) là một dự án mã nguồn mở nhằm hỗ trợ tính toán số với Python và là một thư viện toán học phổ biến và mạnh mẽ của Python. Cho phép làm việc hiệu quả với ma trận và mảng, đặc biệt là dữ liệu ma trận và mảng lớn với tốc độ xử lý nhanh hơn nhiều lần khi chỉ sử dụng “core Python” đơn thuần.

Năm 2005, Travis Oliphant đã tạo NumPy bằng cách đưa các tính năng của mô-đun Numeric vào một mô-đun khác Numarray. Đây là một mô-đun mở rộng của Python, phần lớn được viết bằng C, cung cấp các chức năng khác nhau có khả năng thực hiện các phép tính số với tốc độ cao. Ngoài ra thư viện Numpy còn cung cấp nhiều cấu trúc dữ liệu mạnh mẽ khác nhau, triển khai các mảng và ma trận đa chiều. Các cấu trúc dữ liệu này được sử dụng để tính toán tối ưu thuật toán liên quan đến mảng và ma trận.

Với cuộc cách mạng của khoa học dữ liệu, các thư viện phân tích dữ liệu như NumPy, SciPy, Pandas, … đã có sự phát triển vượt bậc với cú pháp dễ dàng hơn nhiều so với các ngôn ngữ lập trình khác, python là ngôn ngữ lựa chọn hàng đầu cho các nhà khoa học dữ liệu. Thư viện NumPy cung cấp một cách thuận tiện và hiệu quả để xử lý một lượng lớn dữ liệu và cũng rất tiện lợi với phép nhân ma trận và định hình lại dữ liệu, vì thế nó nhanh chóng nên hợp lý khi làm việc với một bộ dữ liệu lớn. Thư viện Numpy có những ưu điểm khi sử dụng để phân tích dữ liệu như sau:

* NumPy thực hiện toán học và logic mảng.
* NumPy thực hiện hiệu quả các mảng đa chiều.
* NumPy thực hiện các phép tính khoa học.
* NumPy có khả năng thực hiện Fourier Transform và định hình lại dữ liệu được lưu trữ trong các mảng đa chiều.
* NumPy cung cấp các hàm tích hợp cho đại số tuyến tính và tạo số ngẫu nhiên.

Sự kết hợp của NumPy với SciPy và Matplotlib được sử dụng thay thế cho MATLAB vì Python là ngôn ngữ lập trình hoàn chỉnh hơn và dễ dàng hơn MATLAB. Các ứng dụng của thư viện NumPy được thể hiện ở Hình 2.2.



Chart, bubble chart

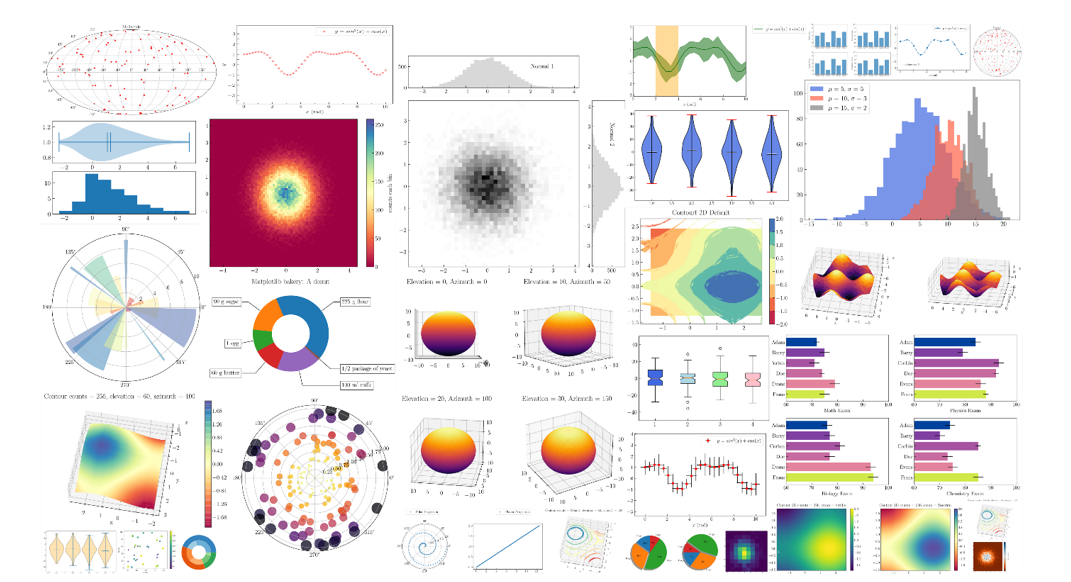
Description automatically generated

Hình 2. 2: Các ứng dụng của thư viện NumPy

* + - 1. Thư viện Matplotlib

“Matplotlib” là một thư viện vẽ đồ thị rất mạnh mẽ hữu ích cho những người làm việc với Python và NumPy. Module được sử dụng nhiều nhất của Matplotib là Pyplot cung cấp giao diện như MATLAB nhưng thay vào đó, nó sử dụng Python và nó là nguồn mở. Các kiểu biểu đồ của Matplotlib có thể được tìm thấy như trong Hình dưới đây.

* Mục tiêu đơn giản hóa tối đa công việc vẽ biểu đồ qua câu lệnh được cung cấp sẵn.
* Matplotlib hỗ trợ rất nhiều loại biểu đồ, đặc biệt là các loại được sử dụng trong nghiên cứu hoặc kinh tế như biểu đồ dòng, đường, tần suất (histograms), phổ, tương quan, errorcharts, scatterplots,…
* Cấu trúc matplotlib gồm nhiều phần được phục vụ cho các mục đích sử dụng khác nhau.



Hình 2. 3: Các biểu đồ trong thư viện Matplotlib

Trong đề tài này, thư viện Matplotlib được sử chủ yếu để biểu diễn kết quả đánh giá mô hình dưới dạng biểu đồ trực quan, dựa vào đó, ta sẽ thực hiện được các bước điều chỉnh các tham số cũng như cấu trúc của mô hình sao cho nó đặt được hiệu năng tối ưu nhất.

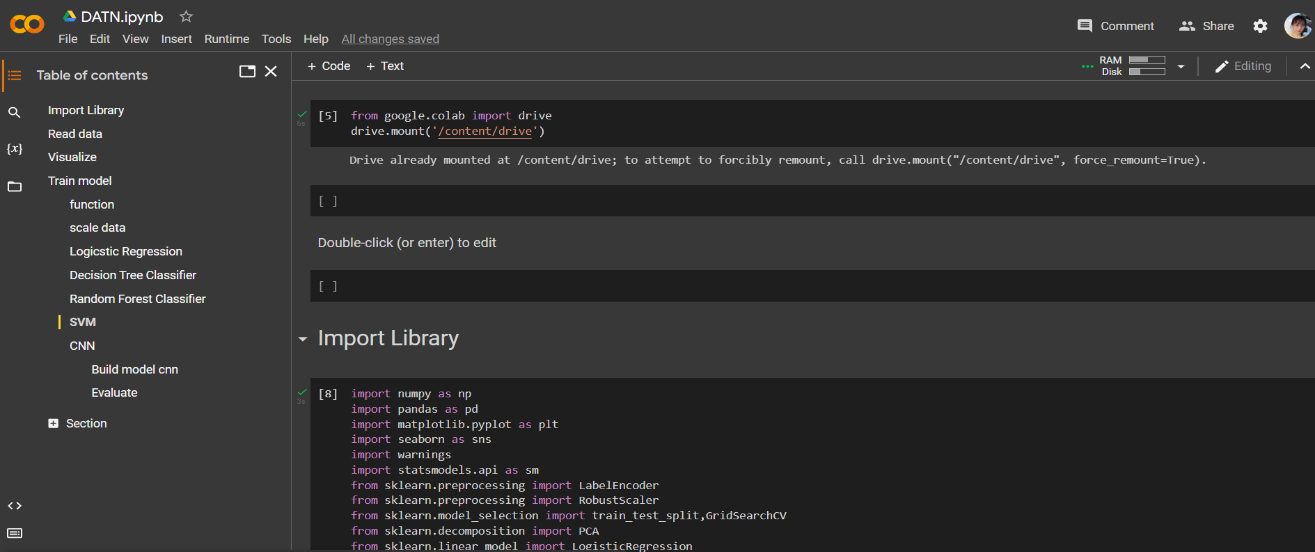
* + - 1. Thư viện Pandas

Thư viện pandas trong python là một thư viện mã nguồn mở, hỗ trợ đắc lực trong thao tác dữ liệu. Đây cũng là bộ công cụ phân tích và xử lý dữ liệu mạnh mẽ của ngôn ngữ lập trình python. Thư viện này được sử dụng rộng rãi trong cả nghiên cứu lẫn phát triển các ứng dụng về khoa học dữ liệu. Thư viện này sử dụng một cấu trúc dữ liệu riêng là Dataframe. Pandas cung cấp rất nhiều chức năng xử lý và làm việc trên cấu trúc dữ liệu này. Chính sự linh hoạt và hiệu quả đã khiến cho pandas được sử dụng rộng rãi.

* DataFrame đem lại sự linh hoạt và hiệu quả trong thao tác dữ liệu và lập chỉ mục;
* Là một công cụ cho phép đọc/ ghi dữ liệu giữa bộ nhớ và nhiều định dạng file: csv, text, excel, sql database, hdf5;
* Liên kết dữ liệu thông minh, xử lý được trường hợp dữ liệu bị thiếu. Tự động đưa dữ liệu lộn xộn về dạng có cấu trúc;
* Dễ dàng thay đổi bố cục của dữ liệu;
* Tích hợp cơ chế trượt, lập chỉ mục, lấy ra tập con từ tập dữ liệu lớn.
* Có thể thêm, xóa các cột dữ liệu;
* Tập hợp hoặc thay đổi dữ liệu với group by cho phép bạn thực hiện các toán tử trên tập dữ liệu;
* Hiệu quả cao trong trộn và kết hợp các tập dữ liệu;
* Lập chỉ mục theo các chiều của dữ liệu giúp thao tác giữa dữ liệu cao chiều và dữ liệu thấp chiều;
* Tối ưu về hiệu năng;
* Pandas được sử dụng rộng rãi trong cả học thuật và thương mại. Bao gồm thống kê, thương mại, phân tích, quảng cáo,…
  1. **Các công nghệ hỗ trợ**
     1. Colaboratory (Google Colab)

Colaboratory (Google Colab) là một sản phẩm từ Google Research dựa trên Jupyter Notebook cho phép chạy các dòng code python thông qua trình duyệt, đặc biệt phù hợp với Data analysis, ML và giáo dục. Google Colab không cần yêu cầu cài đặt hay cấu hình máy tính, mọi thứ có thể chạy thông qua trình duyệt, người nghiên cứu có thể sử dụng tài nguyên máy tính từ CPU tốc độ cao và cả GPUs và cả TPUs đều được cung cấp đầy đủ.

Colab cung cấp nhiều loại GPU và thay đổi theo thời gian. Vì là dịch vụ miễn phí, nên Colab sẽ có những thứ tự ưu tiên trong việc sử dụng tài nguyên hệ thống, cũng như giới hạn thời gian sử dụng, thời gian sử dụng tối đa tới 12 giờ.



Hình 2. 4: Sử dụng google Colab để tạo mô hình

Trong đề tài này, em sử dụng Google Colab làm môi trường lập trình chính để đào tạo mô hình cũng như kiểm thử kết quả huấn luyện bởi sức mạnh đến từ GPU của Google Colab.

* + 1. Kaggle

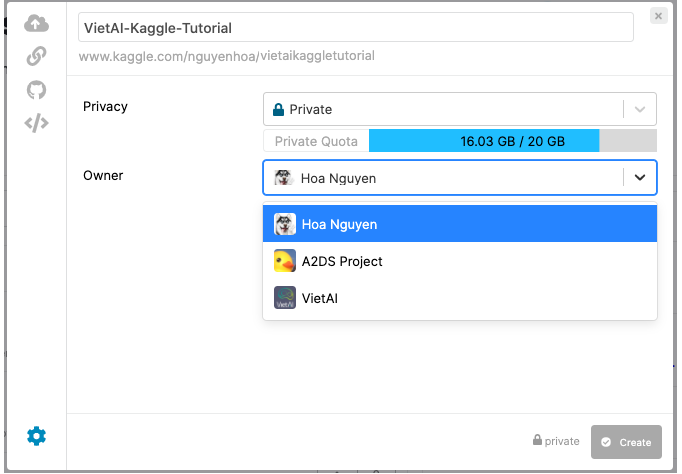
Kaggle là một nền tảng trực tuyến cho cộng đồng Machine Learning (ML) và Khoa học dữ liệu. Kaggle cho phép người dùng chia sẻ, tìm kiếm các bộ dữ liệu; tìm hiểu và xây dựng models, tương tác với những nhà khoa học và kỹ sư ML trên toàn thế giới; tham gia các cuộc thi để có cơ hội chiến thắng những giải thưởng giá trị. Người dùng Kaggle sẽ được hỗ trợ Graphic Processing Unit (GPU) và gần đây có thêm Tensor Processing Unit (TPU) để tăng tốc độ tính toán trong quá trình training cũng như inference.

* [GPU](https://www.kaggle.com/docs/efficient-gpu-usage):
* NVIDIA K80 GPUs hỗ trợ tăng tốc độ training của mô hình deep learning lên đến 12.5 lần.
* Quota: 30 giờ/tuần
* [TPU](https://cloud.google.com/tpu/docs/tpus)
* TPUs là sản phẩm của Google để tăng tốc độ làm việc của các mô hình machiine learning, được support ở Tensorflow 2.1 (cả Keras high-level API và models sử dụng training loop tuỳ chỉnh)
* Quota: 30 giờ/tuần và tối đa 3h trong một lần chạy.

Nền tảng Kaggle có những tính năng tiêu biểu có thể khám phá trong Menu Sidebar bên góc trái màn hình.

* Data

Khi nhấn vào thẻ Data trên Menu Sidebar, bạn sẽ thấy Public Dataset, Dataset mà bạn upload lên Kaggle, những bộ data mà bạn yêu thích.  
Để tạo Dataset trên Kaggle, nhấn vào + New Dataset, một giao diện pop-up sẽ hiện lên. Bạn có thể thay chọn Upload **Private** hay **Public** bằng cách nhấn vào biểu thượng Setting ở góc trái dưới cùng của giao diện pop-up. Với mỗi chế độ, bạn có **20GB** để upload dữ liệu để chia sẽ với team, sử dụng để làm input cho Notebook hoặc làm tài nguyên để tổ chức các cuộc thi cho lớp học. Ngoài ra, cũng có thể tuỳ chọn **Owner** cho bộ dataset là hay là các tổ chức mà bạn tham gia.



4 hình thức để tạo Dataset:

* Upload từ máy tính của mình
* Chia sẻ qua đường link
* Chia sẻ từ Github repository
* Embed từ những kernel khác trên Kaggle
  1. **Thuật toán sử dụng**
     1. Thuật toán K-menas Clustering

K-mean clustering là một phương pháp để tìm các cụm và hạt nhân - trung tâm của cụm trong một tập hợp dữ liệu không được gắn nhãn. Người ta chọn số lượng hạt nhân cụm mong muốn phân chẳng hạn như k cụm. Thuật toán K-mean di chuyển lặp đi lặp lại các hạt nhân để giảm thiểu tổng số trong phương sai cụm. Với một tập hợp các hạt nhân ban đầu, thuật toán Kmeans lặp lại hai bước:

* Đối với mỗi hạt nhân, tính toán khoảng cách giữa các training ponit với nó và nếu gần nó hơn -> sẽ gán là cụm của hạt nhân đấy.
* Sau khi phân được cụm như ở bước trên, thì tiếp theo các training point của các cụm tính toán vector trung bình để được vị trí của hạt nhân mới và lặp lại bước trên đến khi không thể thay đổi được vị trí hạt nhân nữa.

Phân cụm có nhiều hữu ích đặc biệt và cực kỳ phổ biến trong ngành khoa học dữ liệu. Trong đó như:

* Phân tích cụm được sử dụng rộng rãi trong nghiên cứu thị trường, nhận dạng mẫu, phân tích dữ liệu và xử lý ảnh.
* Phân tích cụm cũng có thể giúp các nhà khoa học dữ liệu khám phá ra các nhóm khác hàng của họ. Và họ có thể mô tả đặc điểm nhóm khách hàng của mình dựa trên lịch sử mua hàng.
* Trong lĩnh vực sinh học, nó có thể được sử dụng để xác định phân loại thực vật và động vật, phân loại các gen có chức năng tương tự và hiểu sâu hơn về các cấu trúc vốn có của quần thể.

Vậy để giải các bài toán về phân cụm cần có công cụ/phương pháp, K-means là một trong những thuật toán được sử dụng phổ biến nhất. Trong hướng dẫn này mình sẽ bắt đầu với cơ sở lý thuyết của thuật toán K-means, sau đó sẽ ứng dụng vào 1 ví dụ đơn giản với Python và thư viện Sklearn.

Giả sử muốn chia các điểm sau thành các cụm.



Đầu tiên, phải chọn bao nhiêu cụm mà muốn phân? K trong ‘K-means’ là viết tắt của số cụm mà đang muốn xác định. Mình sẽ bắt đầu với việc chọn 2 cụm, đối với 3, 4 cụm thuật toán cũng sẽ làm tương tự.

Bước thứ 2 là chỉ định 1 điểm, mình sẽ gọi nó là hạt trung tâm để bắt đầu (nó được chọn ngẫu nhiên hoặc cũng được chỉ định bởi các nhà khoa học dữ liệu dựa trên kiến thức trước đó về dữ liệu).

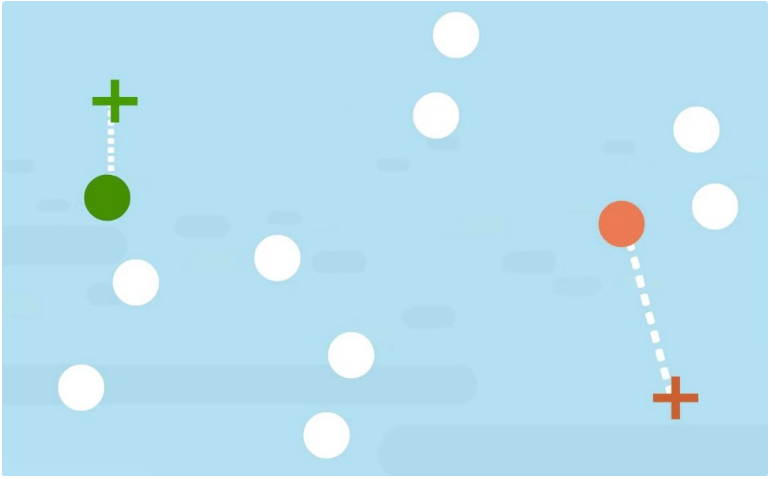
Phân thành 2 cụm sẽ có 2 hạt trung tâm, ở đây là hạt màu xanh lá cây và màu cam như hình dưới.



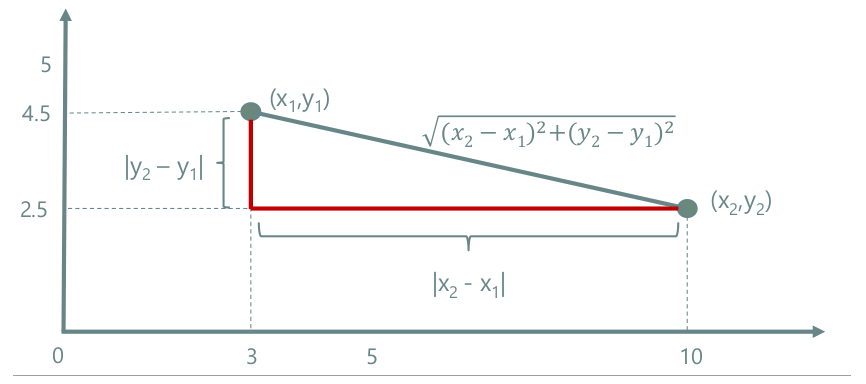
Bước tiếp là tính khoảng cách của hạt trung tâm với các hạt khác. Nếu điểm nào gần trung tâm hơn nó sẽ được gán là màu của hạt trung tâm đó. Ví dụ điểm này gần với hạt màu xanh lá cây hơn là điểm màu cam hơn nên nó sẽ thuộc cụm xanh lá.



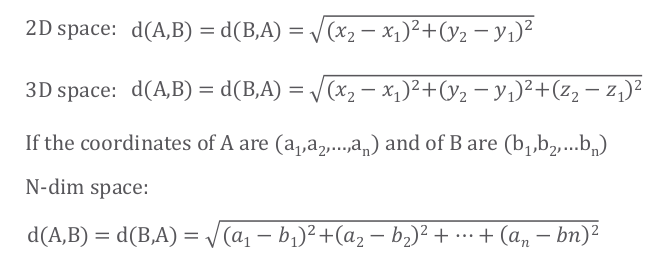
Mặt khác, điểm gần với hạt màu cam hơn nên nó sẽ là 1 phần của cụm cam.



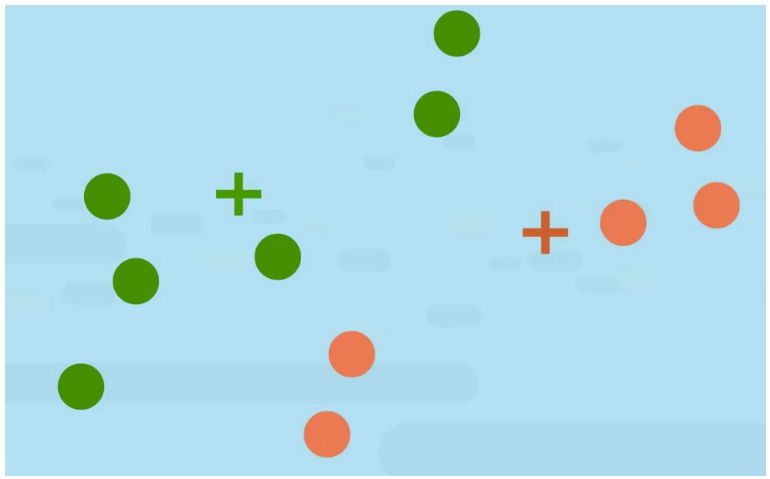
Bằng cách này có thể gán tất cả các điểm trên biểu đồ trong mỗi cụm, dựa trên tính khoảng cách giữa 2 điểm biết tọa độ, dưới đây là công thức nhắc lại? Các bạn có thấy quen quen không



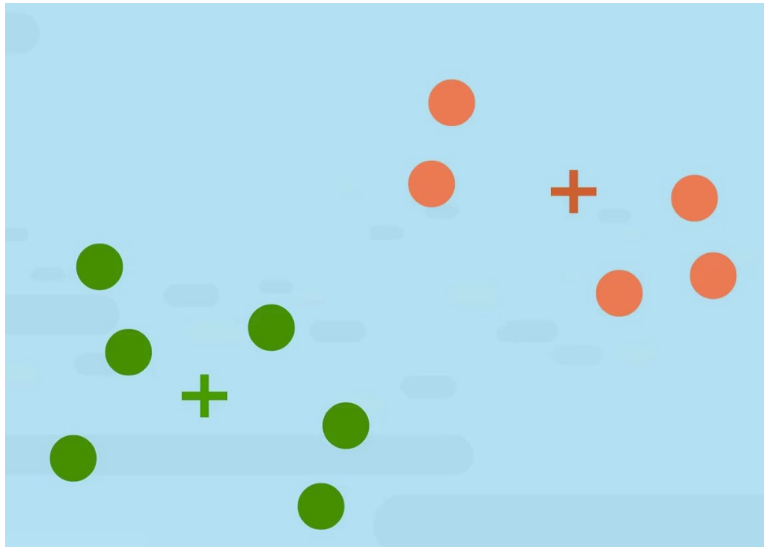
Đây là không gian 2 chiều. Thế đối với không gian 3 chiều hay nhiều chiều thì sao? Dưới đây là công thức tổng quát:



Tiếp tục với biểu đồ phân lớp bên trên sau khi đã gán hết cụm của các điểm, sẽ tính lại tọa độ hạt trung tâm thêm lần nữa rồi lặp lại các bước tính khoảng cách trên để gán lại các điểm.



Tính tọa điểm của hạt trung tâm mới sẽ bằng tổng tọa độ các điểm rồi chia cho số điểm tương ứng - toán học lớp 12. lặp lại sau khoảng 2 lần 3 lần hoặc có thể 1000 lần đến khi đạt được giải pháp phân cụm và không thể điều chỉnh hạt trung tâm được nữa.



* + 1. Thuật toán Decision Tree

Cây quyết định (Decision Tree) là một cây phân cấp có cấu trúc được dùng để phân lớp các đối tượng dựa vào dãy các luật. Các thuộc tính của đối tượngncó thể thuộc các kiểu dữ liệu khác nhau như Nhị phân (Binary) , Định danh (Nominal), Thứ tự (Ordinal), Số lượng (Quantitative) trong khi đó thuộc tính phân lớp phải có kiểu dữ liệu là Binary hoặc Ordinal.

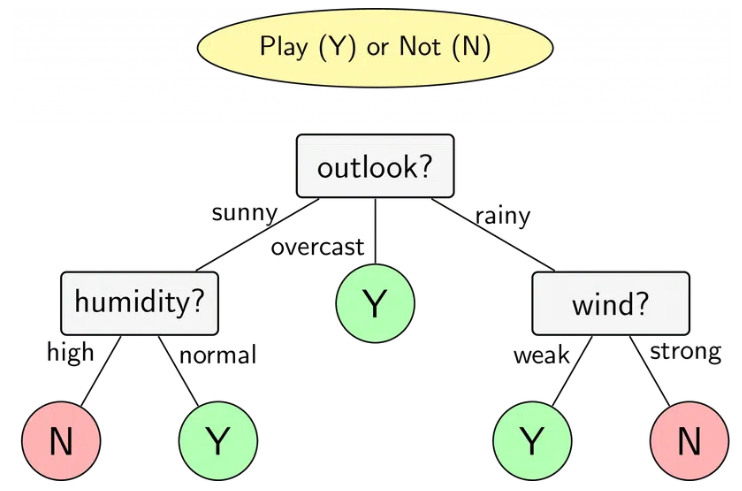
Tóm lại, cho dữ liệu về các đối tượng gồm các thuộc tính cùng với lớp (classes) của nó, cây quyết định sẽ sinh ra các luật để dự đoán lớp của các dữ liệu chưa biết.

Hãy xét một ví dụ 1 kinh điển khác về cây quyết định. Giả sử dựa theo thời tiết mà các bạn nam sẽ quyết định đi đá bóng hay không?

Những đặc điểm ban đầu là:

* Thời tiết
* Độ ẩm
* Gió

Dựa vào những thông tin trên, bạn có thể xây dựng được mô hình như sau:



Dựa theo mô hình trên, thấy:

Nếu trời nắng, độ ẩm bình thường thì khả năng các bạn nam đi chơi bóng sẽ cao. Còn nếu trời nắng, độ ẩm cao thì khả năng các bạn nam sẽ không đi chơi bóng.

# CHƯƠNG 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

* 1. **Phương pháp cài đặt thử nghiệm**

Để tiến hành quá trình thực hành và xây dựng hệ thống và huấn luyện em sử dụng máy core i5-1035G1, Ram 8G, ổ cứng SSD 256GB. . Ngoài ra em sử dụng ngôn ngữ lập trình Python làm ngôn ngữ để xây dựng hệ thống. Trong ngôn ngữ python, có hỗ trợ các thư viện dành cho deep learning như: Tensorflow, sklearn,…, các thư viện thao tác dữ liệu như pandas, numpy, .. và em sử dụng PyQt để xây dựng ứng dụng.

Tensorflow là một thư viện mã nguồn mở cung cấp khả năng xử lý tính toán số học dựa trên biểu đồ mô tả sự thay đổi của dữ liệu. Trong đó, các node là các phép tính toán học còn các cạnh biểu thị luồng dữ liệu. Tensor là cấu trúc dữ liệu trong tensorflow đại diện cho tất cả các loại dữ liệu. Nói cách khác, tất cả các kiểu dữ liệu khi đưa vào trong Tensorflow thì đều được gọi là Tensor. Do vậy, có thể hiểu được Tensorflow là một thư viện mô tả, điều chỉnh dòng chảy của các Tensor. Tensor có 03 thuộc tính cơ bản: là rank, shape và type. Rank là số bậc của tensor. Việc phân rank này khá quan trọng vì nó đồng thời cũng giúp phân loại dữ liệu của Tensor. Shape là chiều của Tensor. Kiểu dữ liệu của các phần tử (elements) trong Tensor. Vì 01 Tensor chỉ có duy nhất 01 thuộc tính Type nên từ đó suy ra chỉ có duy nhất một kiểu Type duy nhất cho toàn bộ phần tử có trong Tensor hiện tại.

Ngoài thư viện Tensorflow, còn sử dụng thư viện Sklearn là thư viện mã nguồn mở hỗ trợ hầu hết các thuật toán của học máy một cách đơn giản mà không cần phải cài đặt, lập trình lại. PyQt là thư viện lập trình giao diện của python được tôi lựa chọn để xây dựng ứng dụng thể hiện quá trình huấn luyện mạng. Các bước tiến hành thí nghiệm bao gồm 6 bước như sau:

Hiểu rõ về bộ dữ liệu: Đầu tiên, cần đọc và hiểu rõ về bộ dữ liệu, bao gồm thông tin về cấu trúc, định dạng, kích thước và nội dung của nó.

Tiền xử lý dữ liệu: Sau khi hiểu rõ về bộ dữ liệu, cần thực hiện các bước tiền xử lý dữ liệu, bao gồm loại bỏ các dữ liệu không hợp lệ, điền giá trị thiếu, chuyển đổi định dạng dữ liệu và chuẩn hóa dữ liệu.

Khám phá dữ liệu: Sau khi tiền xử lý dữ liệu, cần khám phá dữ liệu để hiểu rõ hơn về các đặc điểm và quan hệ giữa các biến trong dữ liệu. Các công cụ khám phá dữ liệu bao gồm trực quan hóa dữ liệu và phân tích thống kê.

Xây dựng mô hình: Sau khi đã hiểu rõ về dữ liệu, cần xây dựng một mô hình dự báo thời tiết. Có thể sử dụng nhiều loại mô hình, bao gồm mô hình hồi quy tuyến tính, mô hình hồi quy logistic, mô hình học sâu và mô hình học máy khác.

Đánh giá và tinh chỉnh mô hình: Sau khi đã xây dựng mô hình, cần đánh giá mô hình bằng các độ đo chất lượng mô hình như độ chính xác, độ phân loại, độ lỗi và độ giải thích. Nếu cần thiết, cần tinh chỉnh mô hình để cải thiện độ chính xác và độ phân loại.

Dự báo và giải thích kết quả: Cuối cùng, cần sử dụng mô hình để dự báo thời tiết và giải thích kết quả. Các công cụ hỗ trợ dự báo thời tiết bao gồm trực quan hóa dữ liệu, biểu đồ và báo cáo.

* 1. **Giới thiệu bộ dữ liệu**

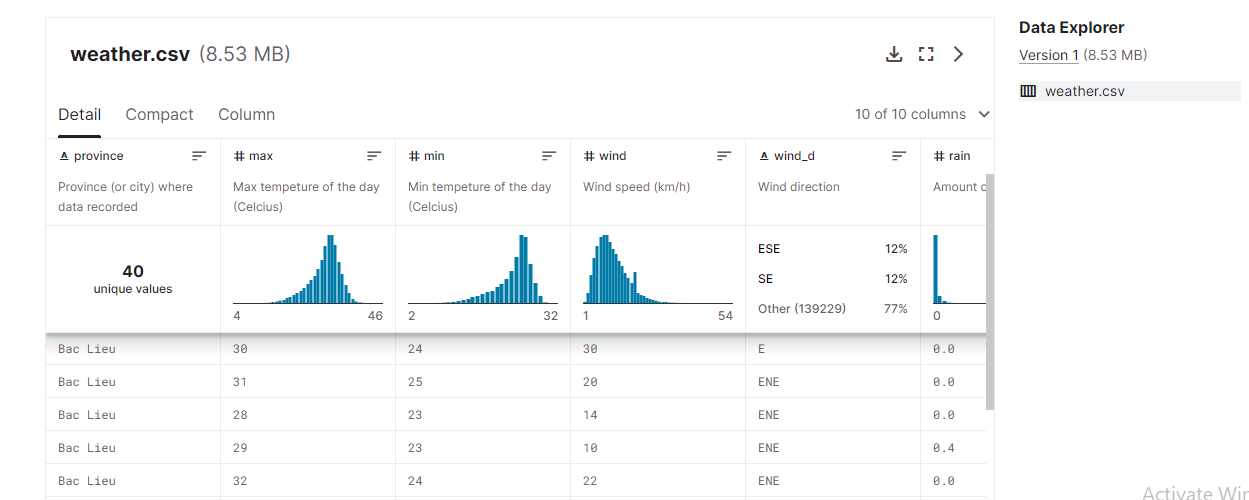
Bộ dữ liệu "Indian Weather Analysis and Forecast" là một bộ dữ liệu về thời tiết được đăng trên nền tảng Kaggle bởi Trivago. Bộ dữ liệu này cung cấp thông tin về dự báo thời tiết hàng ngày của các thành phố ở Ấn Độ từ năm 1901 đến năm 2017. Đây là một bộ dữ liệu lớn và phù hợp để sử dụng cho các mô hình dự báo thời tiết và các nghiên cứu liên quan đến thời tiết và khí hậu ở Ấn Độ.

Bộ dữ liệu "Indian Weather Analysis and Forecast" được cung cấp dưới dạng tập tin CSV với kích thước khoảng 300MB. Bộ dữ liệu chứa thông tin về các thành phố khác nhau ở Ấn Độ, bao gồm Mumbai, Delhi, Bangalore, Hyderabad và Kolkata, và mỗi thành phố có khoảng 5 năm dữ liệu.

Dữ liệu bao gồm các thông tin về nhiệt độ, độ ẩm, tốc độ gió, áp suất không khí và các thông tin khác về thời tiết. Ngoài ra, bộ dữ liệu cũng cung cấp thông tin về tình trạng thời tiết hiện tại, như mưa, tuyết và sương mù. Thông tin trong bộ dữ liệu được cập nhật hàng ngày.

Các cột dữ liệu trong bộ dữ liệu bao gồm:

|  |  |
| --- | --- |
| datetime\_utc | Ngày giờ của thông tin thời tiết |
| conds | Tình trạng thời tiết hiện tại |
| dewptm | Điểm sương hiện tại (độ C) |
| fog | Có mù sương hay không |
| hail | Có mưa đá hay không |
| heatindexm | Chỉ số nhiệt độ hiện tại (độ C) |
| hum | Độ ẩm tương đối (%) |
| precipm | Lượng mưa tích lũy (mm) |
| pressurem | Áp suất không khí hiện tại (hPa) |
| rain | Có mưa hay không |
| snow | Có tuyết hay không |
| tempm | Nhiệt độ hiện tại (độ C) |
| thunder | Có sấm sét hay không |
| tornado | Có lốc xoáy hay không |
| vism | Tầm nhìn hiện tại (km) |
| wdird | Hướng gió hiện tại (độ) |
| wspdm | Tốc độ gió hiện tại (km/h) |
|  |  |

**

Hình 3. 1: Bộ dữ liệu thực nghiệm

Một vài đặc điểm của bộ dữ liệu:

* Bộ dữ liệu chứa 989,232 dòng và 20 cột dữ liệu
* Bộ dữ liệu chứa dữ liệu từ năm 1901 đến 2017

Bộ dữ liệu bao gồm dữ liệu hàng ngày cho năm 1901 đến 2017 và dữ liệu cho tháng đầu tiên của năm 2017.

Dữ liệu của các thành phố được chia thành các tệp riêng biệt, với mỗi tệp đại diện cho một thành phố.

Dữ liệu bị thiếu ở một số cột, ví dụ như tốc độ gió, áp suất không khí và tầm nhìn. Tuy nhiên, các cột chứa thông tin quan trọng như nhiệt độ, độ ẩm và tình trạng thời tiết được cung cấp đầy đủ.

Bộ dữ liệu không có thông tin về vị trí địa lý của các thành phố.

Các ứng dụng của bộ dữ liệu "Indian Weather Analysis and Forecast" có thể là:

Dự báo thời tiết: Bộ dữ liệu này có thể được sử dụng để huấn luyện các mô hình dự báo thời tiết cho các thành phố ở Ấn Độ. Các mô hình này có thể được sử dụng để cung cấp dự báo thời tiết chính xác và đáng tin cậy cho các hoạt động hàng ngày.

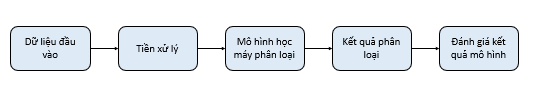
Nghiên cứu khí hậu: Bộ dữ liệu này có thể được sử dụng để nghiên cứu về khí hậu và biến đổi khí hậu ở Ấn Độ. Dữ liệu về nhiệt độ, độ ẩm và áp suất không khí có thể được sử dụng để phân tích xu hướng thay đổi khí hậu và ảnh hưởng của các yếu tố khác nhau đến khí hậu ở Ấn Độ.

Phân tích thị trường: Bộ dữ liệu này cũng có thể được sử dụng để phân tích tác động của thời tiết đến các ngành kinh tế khác nhau ở Ấn Độ, ví dụ như nông nghiệp và du lịch.

Mô hình hóa: Bộ dữ liệu này có thể được sử dụng để huấn luyện các mô hình học máy khác nhau, chẳng hạn như mô hình dự báo thời tiết, mô hình dự báo lượng mưa và mô hình dự báo độ ẩm. Những mô hình này có thể được sử dụng để giải quyết các vấn đề liên quan đến thời tiết và khí hậu ở Ấn Độ.

* 1. **Quá trình phân tích thực nghiệm**

Các bước phân tích thực nghiệm được thể hiện như sơ đồ sau:



Hình 3. 2: Sơ đồ phân tích thực nghiệm

Đây là quá trình phân tích thực nghiệm cho một bài toán phân loại thời tiết sử dụng bộ dữ liệu "Indian Weather Analysis and Forecast" trên Kaggle:

Dữ liệu đầu vào: Đầu tiên, phải thu thập và tải bộ dữ liệu "Indian Weather Analysis and Forecast" từ Kaggle. Sau đó, phải tiến hành khám phá dữ liệu để hiểu rõ về cấu trúc và nội dung của nó.

Tiền xử lý: Sau khi đã hiểu rõ về dữ liệu, cần tiền xử lý dữ liệu trước khi đưa vào mô hình học máy. Các bước tiền xử lý có thể bao gồm: loại bỏ các dữ liệu không hợp lệ, xử lý giá trị thiếu, chuẩn hóa dữ liệu và chuyển đổi định dạng dữ liệu.

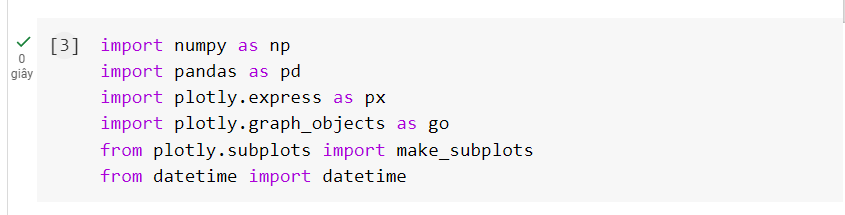
Mô hình học máy phân loại: Sau khi đã tiền xử lý dữ liệu, có thể sử dụng các mô hình học máy phân loại để dự đoán thời tiết. Các mô hình học máy phân loại phổ biến có thể bao gồm: hồi quy logistic, cây quyết định, máy vector hỗ trợ (SVM) và mạng nơ-ron.

Kết quả phân loại: Sau khi đã xây dựng mô hình, có thể sử dụng nó để phân loại các mẫu dữ liệu thời tiết mới. Kết quả phân loại có thể được trực quan hóa bằng các biểu đồ hoặc báo cáo.

Đánh giá kết quả mô hình: Cuối cùng, cần đánh giá kết quả của mô hình để biết được độ chính xác của nó. Các độ đo chất lượng mô hình có thể bao gồm: độ chính xác, độ phân loại, độ lỗi và độ giải thích. Nếu kết quả của mô hình không đạt yêu cầu, có thể tinh chỉnh mô hình hoặc sử dụng các mô hình học máy khác để cải thiện kết quả.

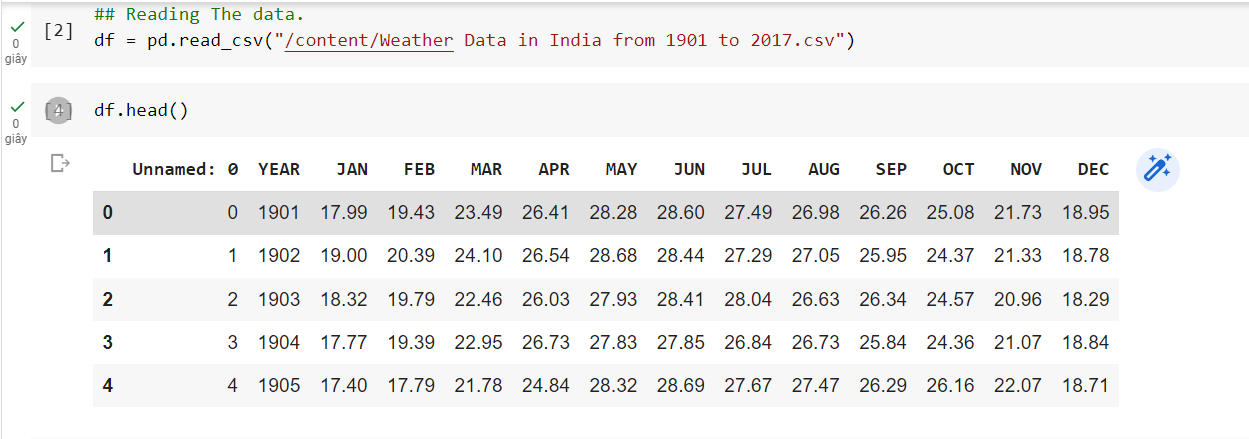
* 1. **Triển khai thực nghiệm**
     1. Cài đặt thư viện và đọc dữ liệu

|  |
| --- |
| import numpy as np  import pandas as pd  import plotly.express as px  import plotly.graph\_objects as go  from plotly.subplots import make\_subplots  from datetime import datetime |

****

Hình 3. 3: Cài đặt thư viện

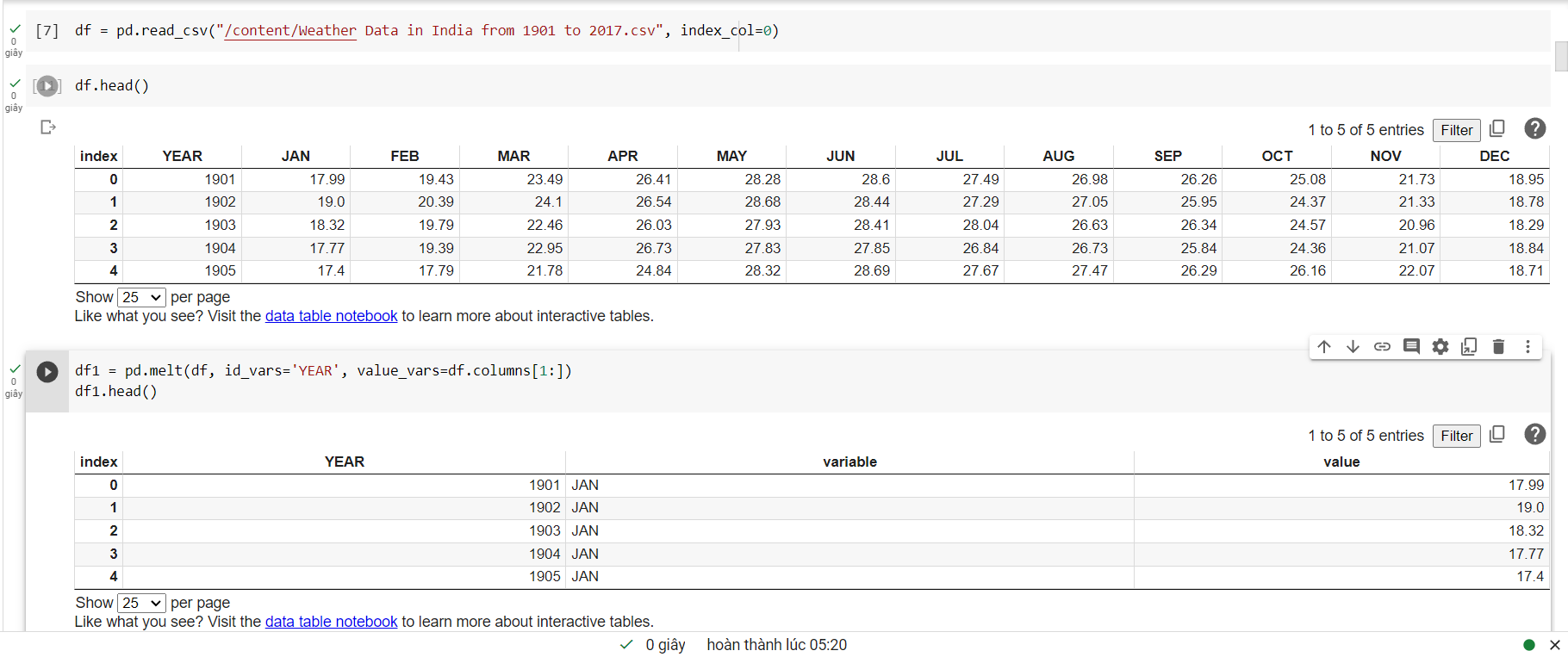
* Đọc dữ liệu



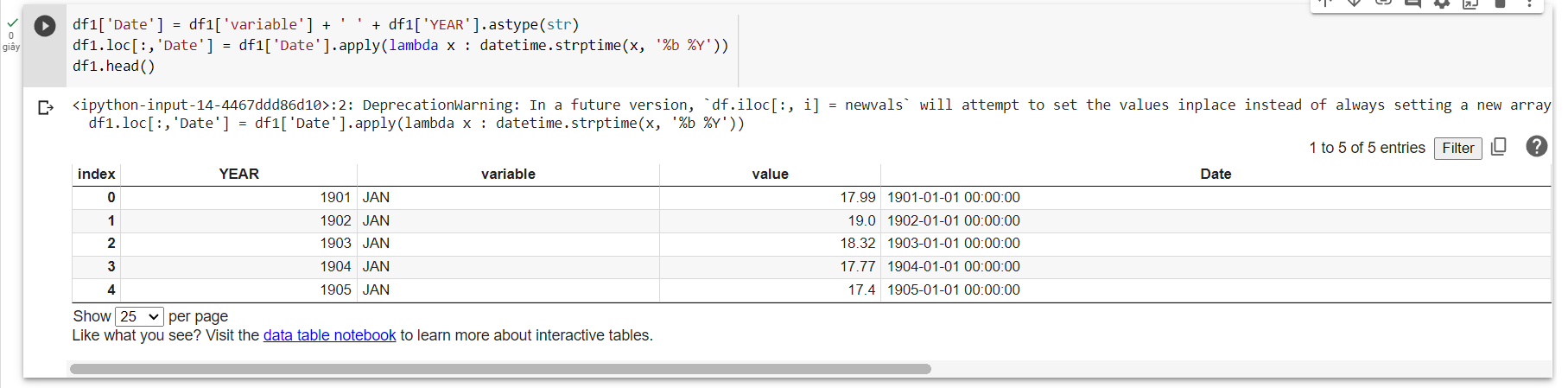
Hình 3. 4: Đọc dữ liệu

* + 1. Mô tả thuộc tính

Mô tả chi tiết từng thuộc tính trong bộ dữ liệu, bao gồm kiểu dữ liệu, đơn vị đo, phân bố giá trị, và ý nghĩa của từng thuộc tính.



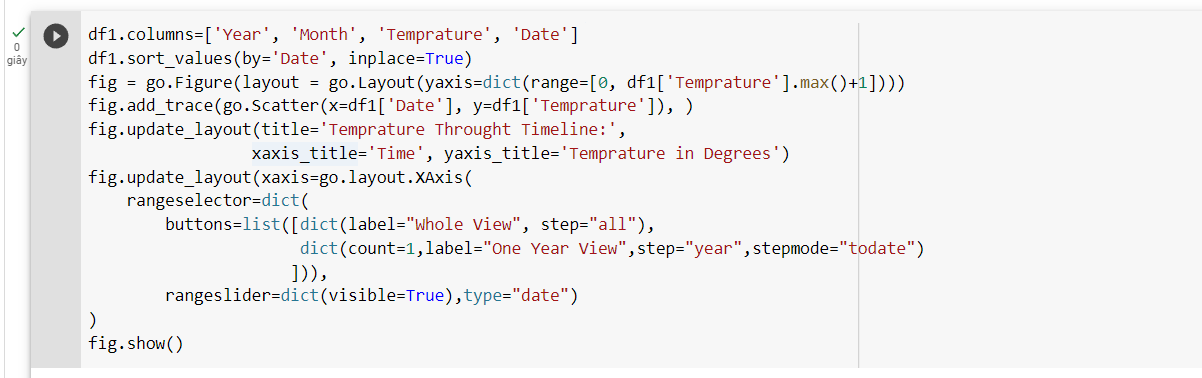
Hình 3. 5: Mô tả thuộc tính

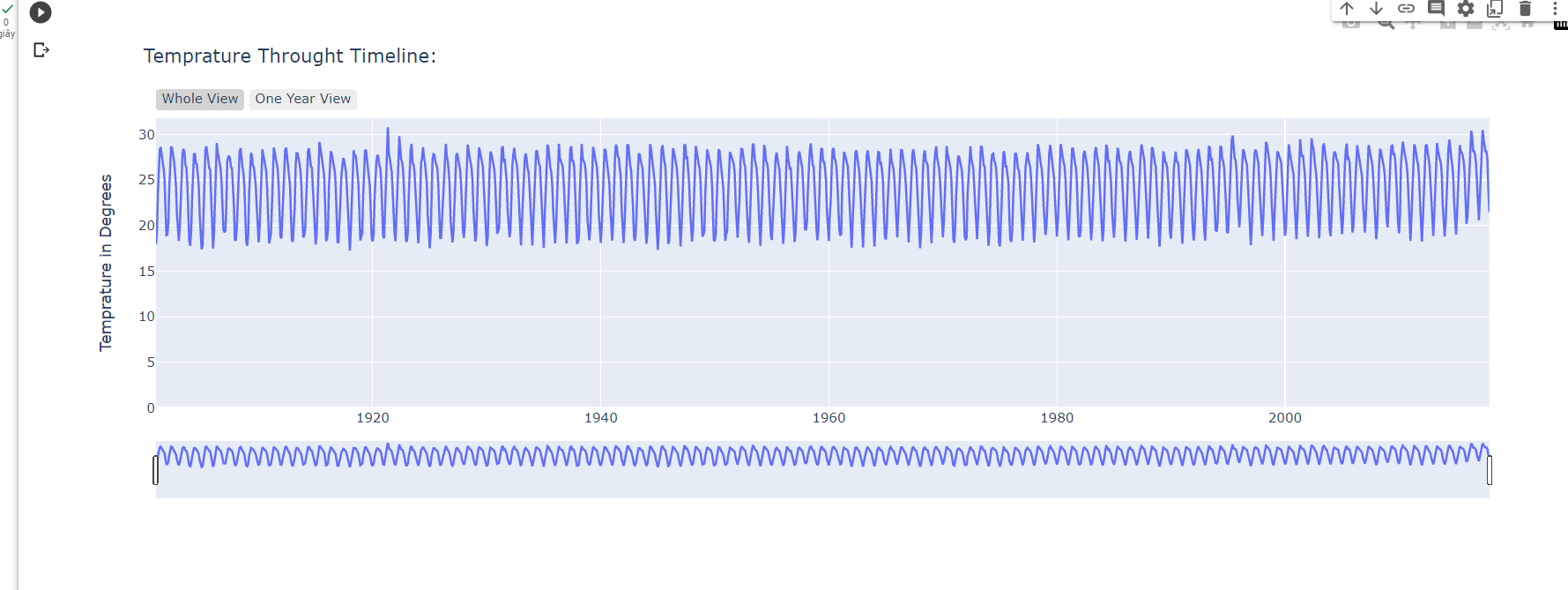


Hình 3. 6: Chuyển đổi chuỗi thành đối tượng datetime

* + 1. Trực quan hóa dữ liệu

Sử dụng các biểu đồ và đồ thị để trực quan hóa dữ liệu và giúp hiểu rõ hơn về các thuộc tính trong bộ dữ liệu.



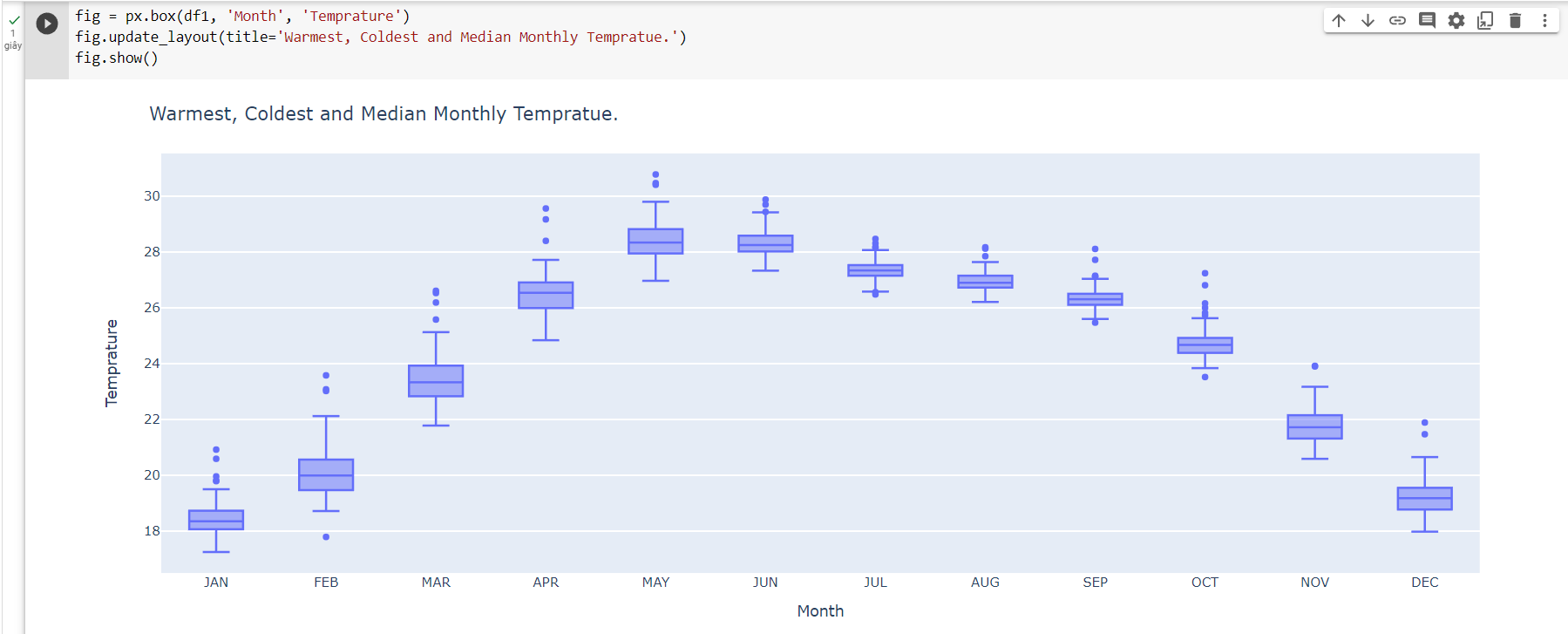


Hình 3. 7: Nhiệt độ xuyên suốt thời gian

Có bốn mùa chính và đây là cách chúng được nhóm lại:

* Mùa đông: tháng 12, tháng 1 và tháng 2.
* Mùa hè (Còn được gọi là "Mùa trước gió mùa") : tháng 3, tháng 4 và tháng 5.
* Gió mùa: Tháng 6, 7, 8 và 9.
* Mùa thu (Còn được gọi là "Mùa hậu gió mùa): tháng 10 và tháng 11.

Đây là cơ sở lý thuyết để bám vào các mùa này để phân tích.

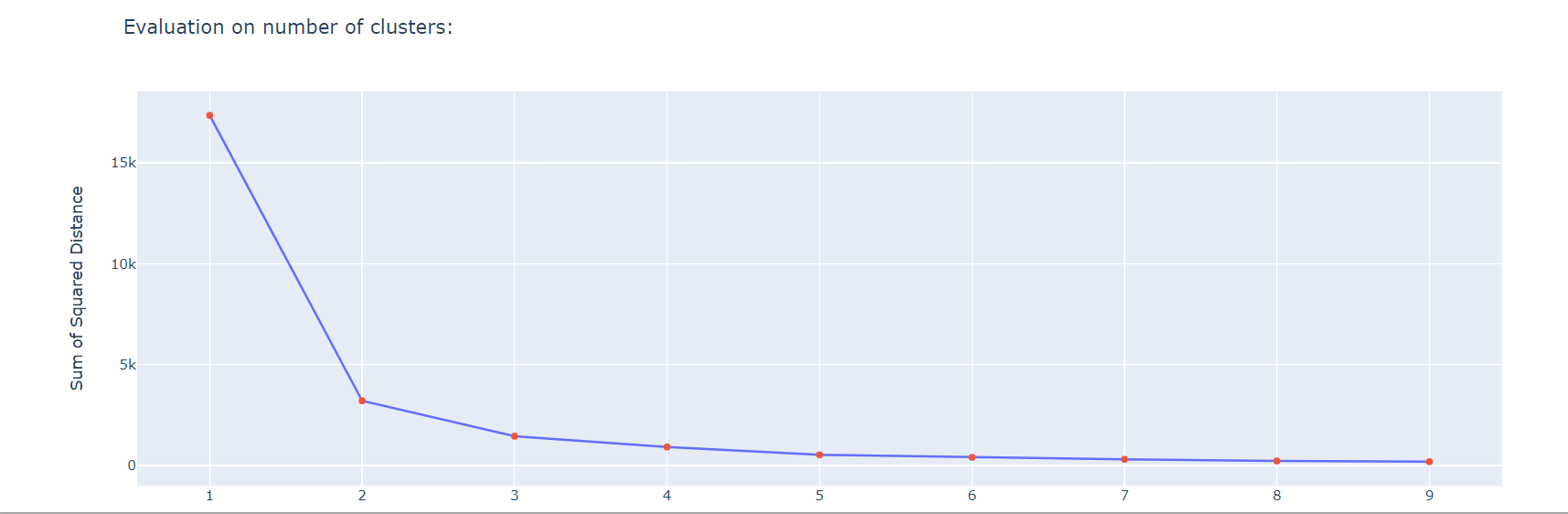


Thông tin chi tiết:

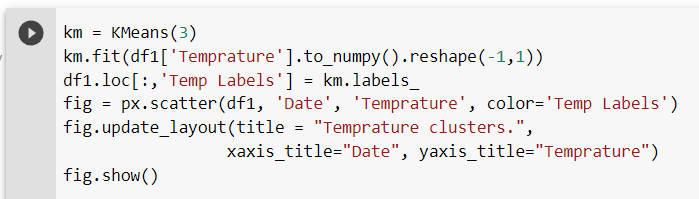
* Tháng giêng có những ngày lạnh nhất trong năm.
* Có thể có những ngày nóng nhất trong một năm.
* Tháng 7 là tháng có độ lệch chuẩn ít nhất, có nghĩa là nhiệt độ tháng 7 thay đổi ít nhất. Có thể mong đợi bất kỳ ngày nào trong tháng bảy là một ngày ấm áp.
  + 1. Xây dựng mô hình

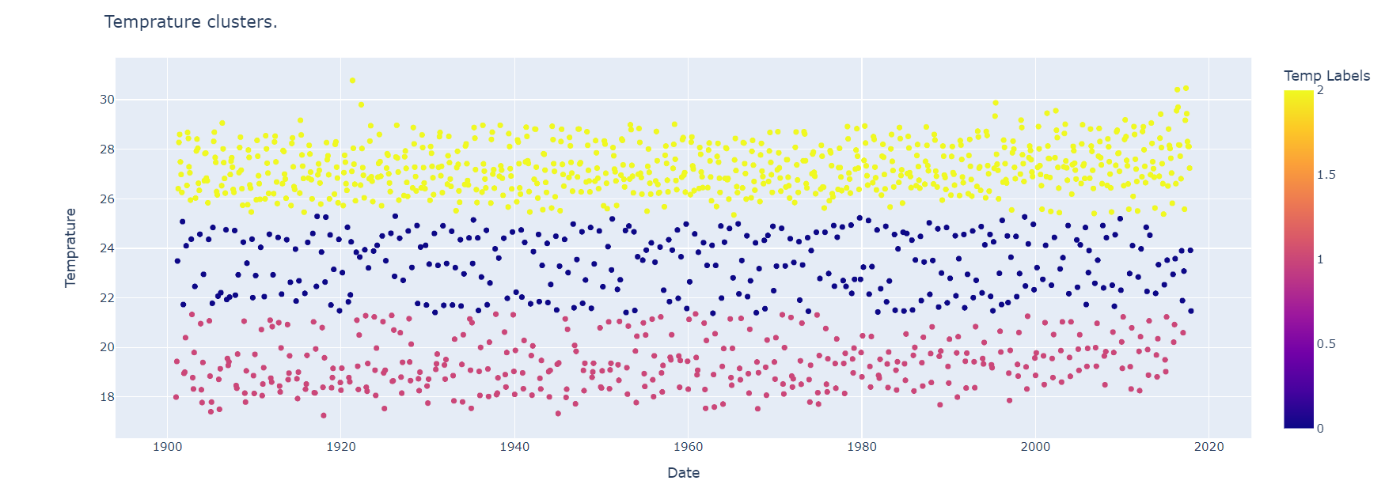
Sử dụng các kỹ thuật học máy để xây dung các mô hình dự đoán trên bộ dữ liệu, bao gồm phân loại, dự báo, và phân tích đối tượng.





Hình 3. 8: Đánh giá V số cụm





Hình 3. 9: Cụm nhiệt độ

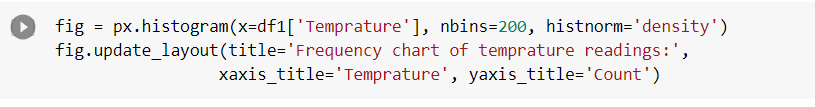
Thông tin chi tiết:

Mặc dù có 4 mùa nhưng có thể thấy 3 cụm chính dựa trên nhiệt độ.

Tháng 1, tháng 2 và tháng 12 là những tháng lạnh nhất.

tháng 4, tháng 5, tháng 6, tháng 7, tháng 8 và tháng 9; tất cả đều có tempratures nóng hơn.

Tháng 3, tháng 10 và tháng 11 là những tháng có nhiệt độ không quá nóng cũng không quá lạnh

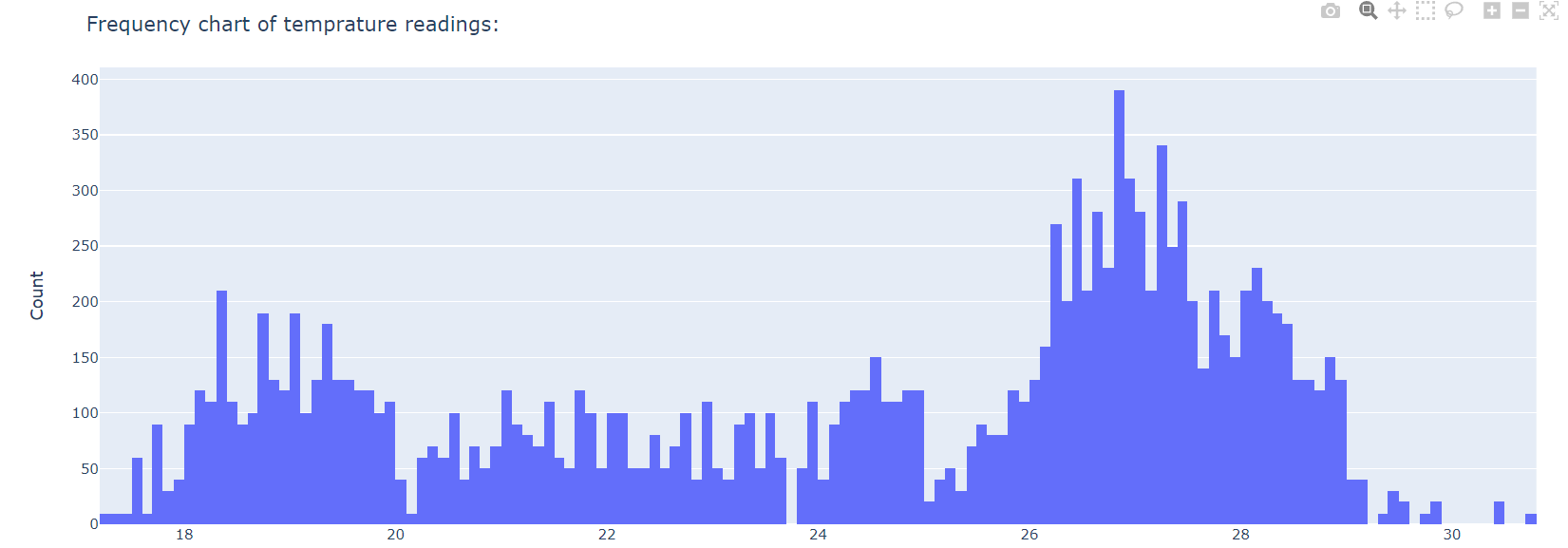


Đoạn code này sử dụng thư viện plotly để tạo biểu đồ histogram cho cột dữ liệu nhiệt độ (Temperature) trong bảng dữ liệu df1.

Cụ thể, đoạn code sử dụng hàm `px.histogram()` để tạo histogram, với đối số `x` là cột dữ liệu nhiệt độ. Tham số `nbins` được sử dụng để chỉ định số lượng các bin (ngăn) trong histogram. Ở đây, số lượng bin được đặt là 200, nghĩa là histogram sẽ chia dữ liệu thành 200 ngăn. Tham số `histnorm` được sử dụng để đặt chế độ chuẩn hóa của histogram. Ở đây, chế độ chuẩn hóa được đặt là 'density', nghĩa là histogram sẽ được chuẩn hóa sao cho tổng diện tích của các cột là 1.

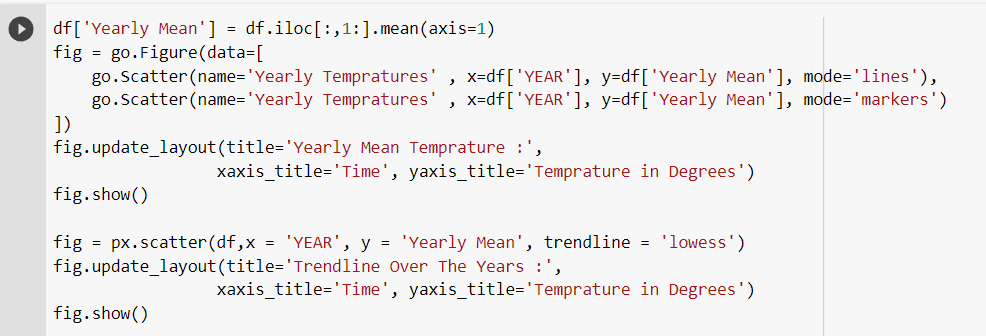
Sau đó, đoạn code sử dụng hàm `update\_layout()` để tạo tiêu đề và đặt tên cho các trục của histogram. Tiêu đề được đặt là 'Frequency chart of temprature readings', trục x được đặt tên là 'Temperature', và trục y được đặt tên là 'Count'.

Kết quả là đoạn code sẽ tạo ra một biểu đồ histogram cho cột dữ liệu nhiệt độ trong bảng dữ liệu df1, hiển thị số lần xuất hiện của các giá trị nhiệt độ trên trục x và tần suất (số lần xuất hiện chia tổng số giá trị) trên trục y. Biểu đồ sẽ giúp nhà khoa học dữ liệu hiểu rõ hơn về phân bố của dữ liệu nhiệt độ và phát hiện ra những đặc điểm thú vị trong dữ liệu.



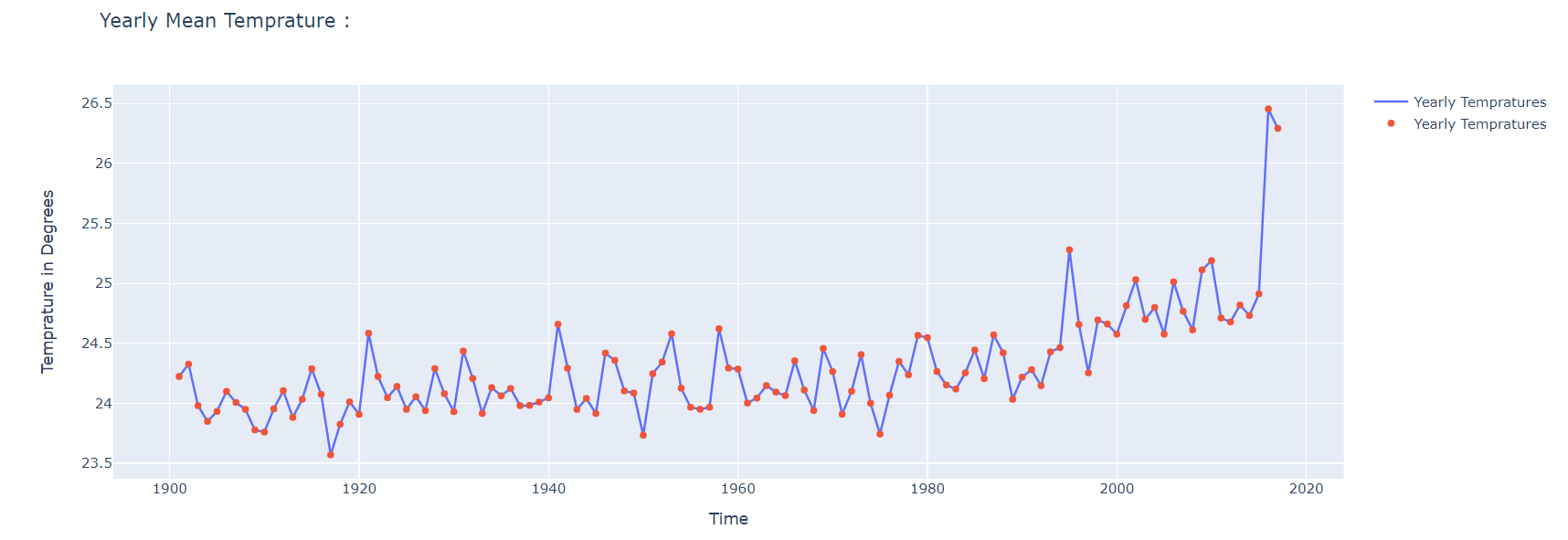
Hình 3. 10: Biểu đồ tần suất nhiệt độ

**Nhiệt độ trung bình hằng năm**

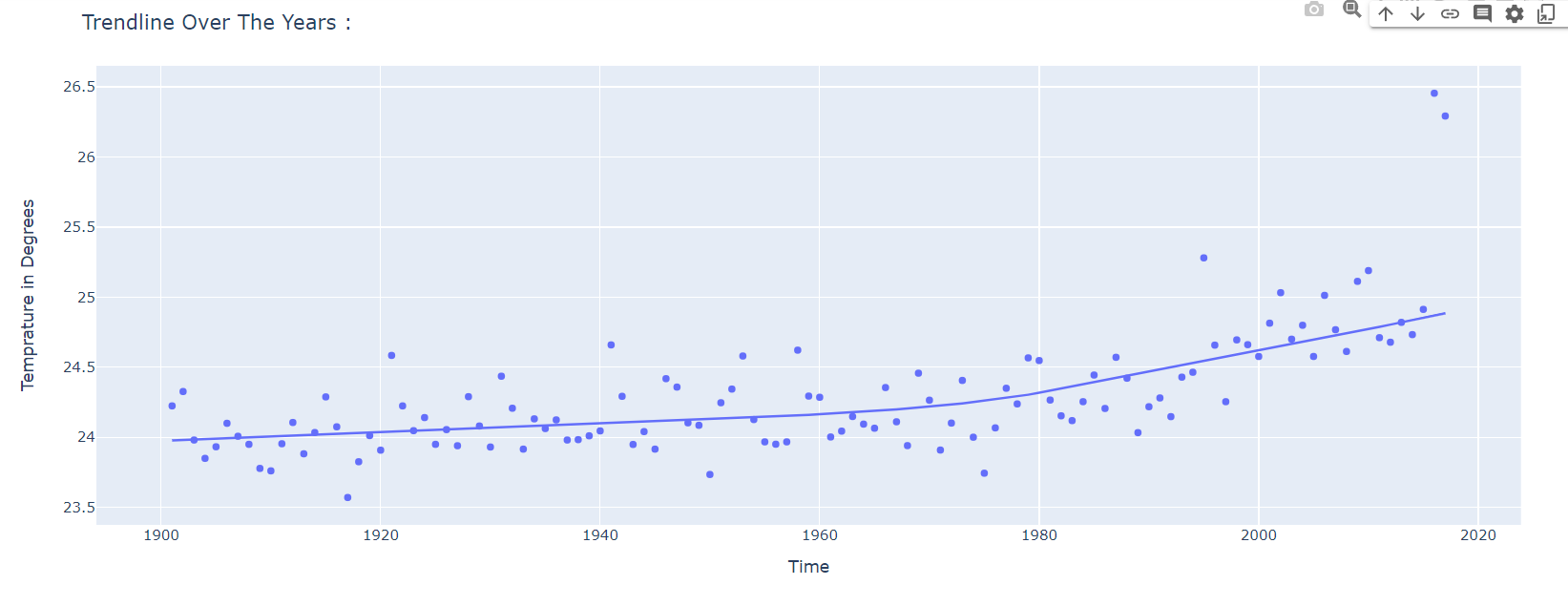


Đoạn code trên tính trung bình nhiệt độ hàng năm (Yearly Mean Temperature) của các thành phố trong dữ liệu, sau đó sử dụng thư viện Plotly để vẽ biểu đồ đường và biểu đồ phân tán.

* Cụ thể, dòng `df['Yearly Mean'] = df.iloc[:,1:].mean(axis=1)` tính toán trung bình cộng của các cột dữ liệu từ cột thứ 2 (cột nhiệt độ) đến cột cuối cùng của DataFrame `df` và lưu kết quả vào cột mới có tên là 'Yearly Mean'.
* Dòng `fig = go.Figure(data=[go.Scatter(name='Yearly Tempratures' , x=df['YEAR'], y=df['Yearly Mean'], mode='lines'), go.Scatter(name='Yearly Tempratures' , x=df['YEAR'], y=df['Yearly Mean'], mode='markers')])` tạo một đối tượng Figure của Plotly với hai đường biểu diễn nhiệt độ trung bình hàng năm. Cụ thể, đường đầu tiên được hiển thị dưới dạng đường thẳng, đường thứ hai được hiển thị dưới dạng điểm, mỗi điểm biểu thị cho một năm trong dữ liệu.
* Dòng `fig.update\_layout(title='Yearly Mean Temprature :', xaxis\_title='Time', yaxis\_title='Temprature in Degrees')` cập nhật tiêu đề và các nhãn trục của biểu đồ đường.
* Dòng `fig.show()` hiển thị biểu đồ đường trong Jupyter notebook hoặc IDE tương tự.
* Dòng `fig = px.scatter(df,x = 'YEAR', y = 'Yearly Mean', trendline = 'lowess')` tạo một đối tượng Figure của Plotly với biểu đồ phân tán và đường xu hướng của nhiệt độ hàng năm. Biểu đồ phân tán hiển thị nhiệt độ hàng năm của các thành phố theo thời gian. Đường xu hướng được xác định bằng phương pháp smoothing LOESS (tức là một phương pháp smoothing dựa trên hồi quy phi tuyến).
* Dòng `fig.update\_layout(title='Trendline Over The Years :', xaxis\_title='Time', yaxis\_title='Temprature in Degrees')` cập nhật tiêu đề và các nhãn trục của biểu đồ phân tán.
* Dòng `fig.show()` hiển thị biểu đồ phân tán trong Jupyter notebook hoặc IDE tương tự.



Hình 3. 11: Nhiệt độ trung bình hàng năm



Hình 3. 12: Đường xu hướng qua các năm

Có thể thấy rằng vấn đề cảnh báo toàn cầu là đúng.

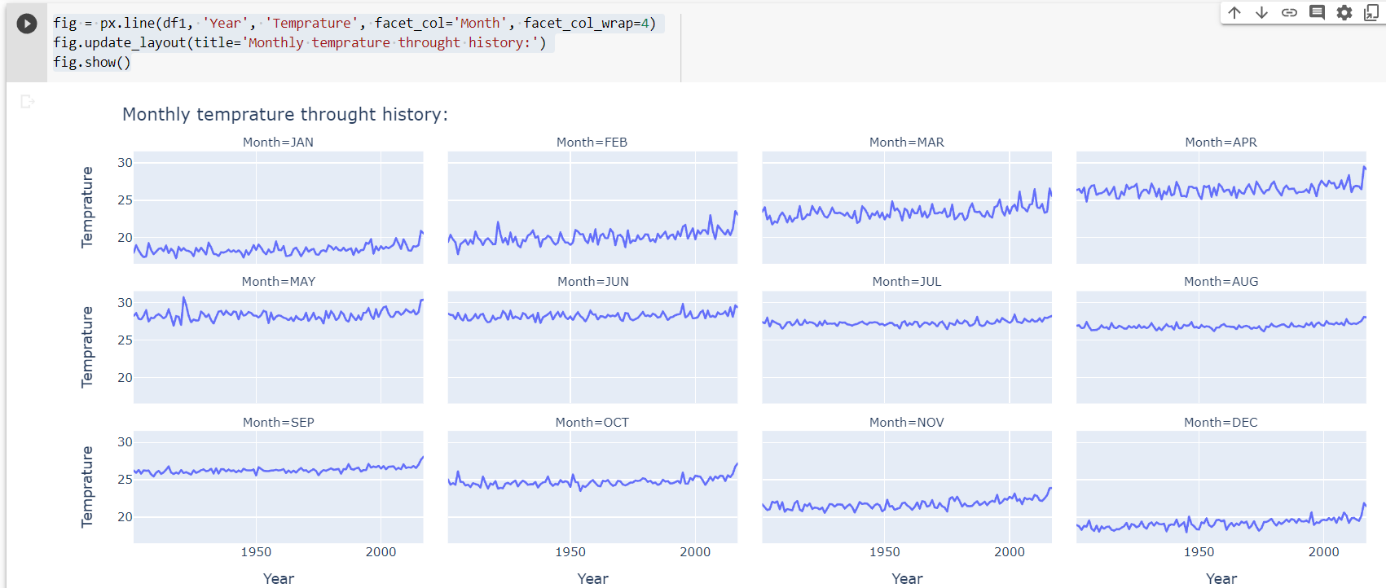
Nhiệt độ trung bình hàng năm không tăng cho đến năm 1980. Chỉ sau năm 1979, mới có thể thấy nhiệt độ trung bình hàng năm tăng dần.

Sau năm 2015, nhiệt độ hàng năm đã tăng mạnh.

Cũng đang thấy một mô hình tăng giảm giống như hàng tháng trong các nhịp độ hàng năm.

Điều này không thể hiểu được. Bởi vì với các tháng, có hiện tượng các mùa và trái đất quay quanh mặt trời theo đường elip. Nhưng mô hình này không được mong đợi trong thời gian hàng năm.

**Nhiệt độ hàng tháng trong suốt lịch sử.**



Đoạn code trên sử dụng thư viện plotly express để tạo ra biểu đồ đường (line chart) với dữ liệu thời tiết theo năm và tháng.

Cụ thể, `px.line()` được sử dụng để tạo đồ thị đường với trục x là năm ('Year') và trục y là nhiệt độ ('Temprature'), trong khi `facet\_col='Month'` cho biết các biểu đồ con sẽ được tạo thành trên cùng một trang, mỗi biểu đồ con hiển thị dữ liệu theo từng tháng, với 4 biểu đồ con hiển thị trên một dòng (`facet\_col\_wrap=4`).

Cuối cùng, `fig.update\_layout()` được sử dụng để cập nhật tiêu đề của biểu đồ.

**Phân tích theo mùa**



Trong đoạn code trên, ta sử dụng các cột dữ liệu mùa đông (DEC, JAN, FEB), mùa hè (MAR, APR, MAY), mùa mưa (JUN, JUL, AUG, SEP), và mùa thu (OCT, NOV) để tính toán nhiệt độ trung bình của từng mùa.

Cụ thể, dùng hàm `mean` của pandas để tính toán nhiệt độ trung bình của từng mùa theo từng hàng của bảng dữ liệu. Kết quả thu được được lưu vào 4 cột mới trong bảng dữ liệu `df`, tương ứng với nhiệt độ trung bình của mỗi mùa.

Sau đó, lấy ra các cột năm và nhiệt độ trung bình của các mùa, và sử dụng hàm `melt` của pandas để chuyển đổi bảng dữ liệu từ dạng rộng sang dạng dài. Kết quả thu được là một bảng dữ liệu mới `seasonal\_df`, trong đó mỗi dòng đại diện cho nhiệt độ trung bình của một mùa trong một năm.

Cuối cùng, vẽ biểu đồ dạng scatter plot sử dụng `px.scatter()` trong plotly, trong đó trục x là năm, trục y là nhiệt độ trung bình của mỗi mùa, và màu sắc của điểm ảnh được dùng để biểu thị cho từng mùa.

*Dự báo và dự đoán ra kết quả*

Dự báo nhiệt độ trung bình hàng tháng cho năm 2023, như sau:



Đoạn code này sử dụng thư viện Scikit-learn để xây dựng một mô hình Decision Tree Regression để dự đoán nhiệt độ. Cụ thể, đoạn code thực hiện các bước sau:

Lựa chọn các thuộc tính của bộ dữ liệu cần sử dụng cho mô hình: 'Year', 'Month' và 'Temperature'.

Áp dụng phương pháp one-hot encoding bằng hàm pd.get\_dummies() để chuyển đổi các thuộc tính dạng chuỗi (Year và Month) thành dạng số.

Tách các giá trị đầu vào (x) và đầu ra (y) từ bộ dữ liệu đã tiền xử lý. Trong đó, giá trị đầu ra là nhiệt độ (Temprature) và giá trị đầu vào là các thuộc tính khác ngoại trừ Temprature.

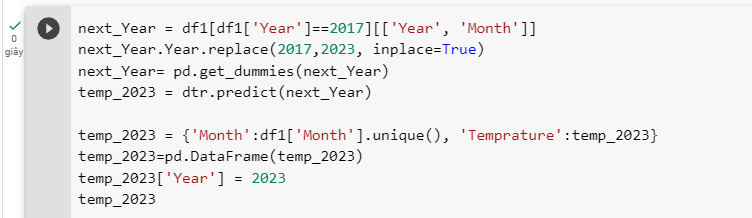
Khởi tạo một mô hình DecisionTreeRegressor() (dtr).

Sử dụng hàm train\_test\_split() để chia bộ dữ liệu thành tập huấn luyện và tập kiểm tra với tỷ lệ 70:30.

Tiến hành huấn luyện mô hình bằng cách gọi phương thức fit() trên tập huấn luyện.

Sử dụng mô hình đã huấn luyện để dự đoán giá trị đầu ra trên tập kiểm tra và tính toán hệ số R2 bằng hàm r2\_score().

Hệ số R2 được sử dụng để đánh giá độ chính xác của mô hình. Nó cho biết tỉ lệ phương sai giữa giá trị dự đoán và giá trị thực tế được giải thích bởi mô hình. Giá trị R2 càng gần 1 thì mô hình càng chính xác.

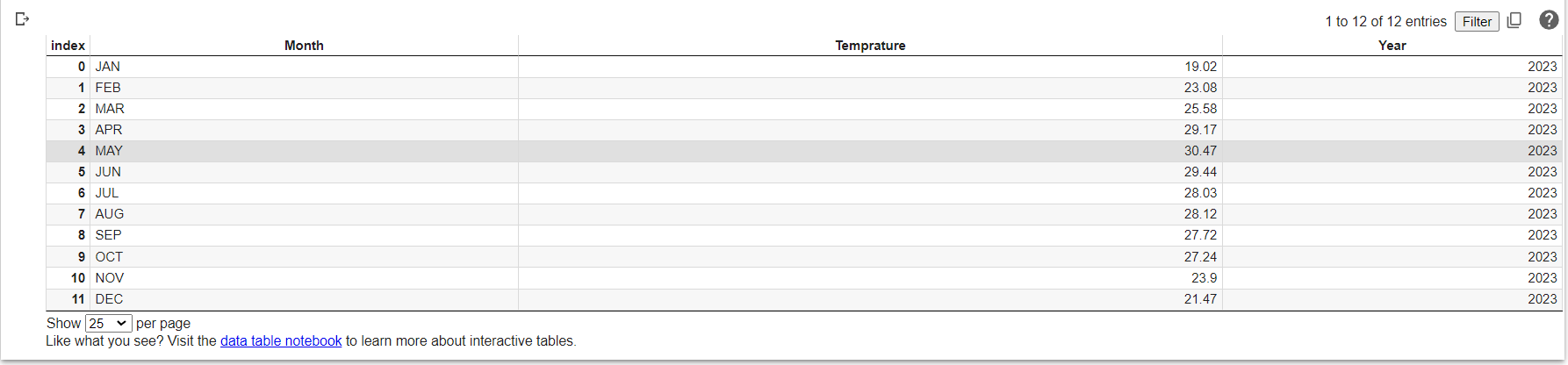


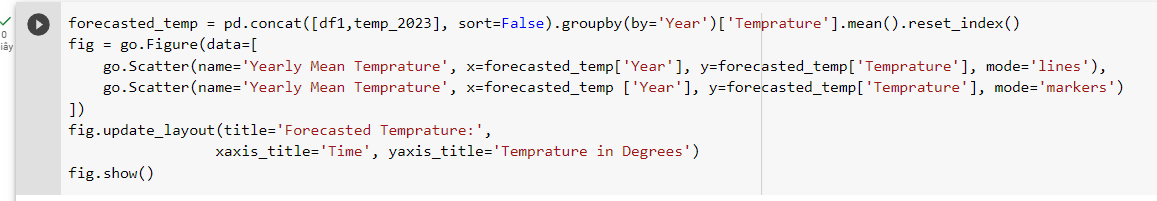
Đoạn code này dùng để dự đoán nhiệt độ cho năm 2023 bằng mô hình hồi quy cây quyết định (DecisionTreeRegressor) được huấn luyện bằng dữ liệu nhiệt độ của các năm trước.

Đầu tiên, ta lọc ra dữ liệu của năm 2017 và chỉ lấy cột 'Year' và 'Month' cho vào biến next\_Year. Sau đó, ta thay đổi giá trị 'Year' của next\_Year từ 2017 thành 2023 bằng cách sử dụng phương thức replace của pandas. Tiếp theo, ta sử dụng pd.get\_dummies để tạo biến giả cho các tháng trong năm 2023.

Sau khi có được dữ liệu cho năm 2023, ta sử dụng mô hình hồi quy cây quyết định đã được huấn luyện để dự đoán nhiệt độ cho từng tháng của năm 2023 và lưu kết quả vào biến temp\_2023.

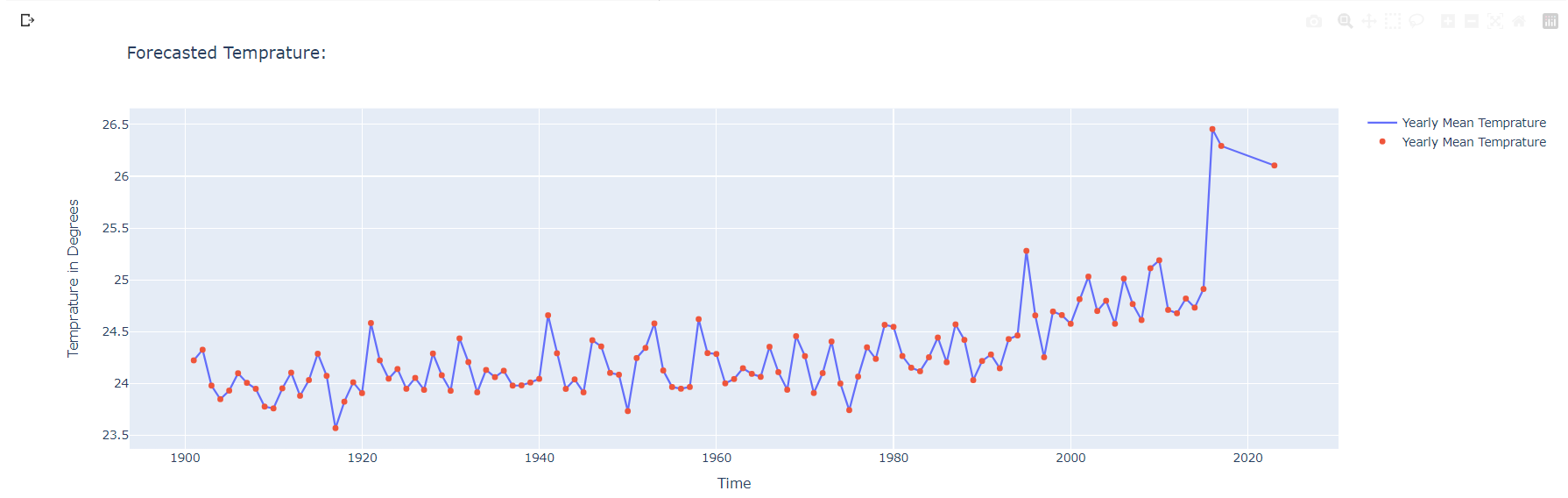
Cuối cùng, ta tạo dataframe mới bao gồm các cột 'Month', 'Temprature' và 'Year' và gán giá trị 'Year' của nó bằng 2023, kết quả dự đoán của nhiệt độ trong năm 2023 được lưu vào cột 'Temprature' của dataframe này.





Đoạn code này được sử dụng để trực quan hóa dữ liệu dự báo nhiệt độ. Đầu tiên, các dữ liệu về nhiệt độ dự báo cho năm 2023 được tính toán bằng cách sử dụng mô hình Decision Tree Regression (DTR) đã được huấn luyện trong đoạn mã trước. Kết quả được lưu trữ trong biến temp\_2023.

Sau đó, ta nối các dữ liệu dự báo với dữ liệu nhiệt độ ban đầu để tính toán nhiệt độ trung bình hàng năm, được lưu trữ trong biến forecasted\_temp. Cuối cùng, ta sử dụng biểu đồ đường (line chart) để hiển thị dữ liệu này, trong đó trục x là thời gian (năm) và trục y là nhiệt độ trung bình hàng năm. Biểu đồ cũng hiển thị các điểm dữ liệu, tương ứng với nhiệt độ trung bình hàng năm tại mỗi năm. Các thông tin khác của biểu đồ được chỉnh sửa thông qua phương thức update\_layout() để đặt tiêu đề và nhãn trục cho biểu đồ.



# KẾT LUẬN

1. **Kết quả đạt được**

* Trong đề tài đồ án tốt nghiệp này, đã thực hiện phân tích dữ liệu thời tiết bằng bộ dữ liệu "Indian Weather Analysis and Forecast". Đã tạo ra nhiều biểu đồ và hình ảnh minh họa để trực quan hóa các xu hướng và mối quan hệ giữa các biến trong dữ liệu.
* Từ phân tích, đã tìm thấy nhiều kết quả kết quả. Cụ thể, thấy rằng nhiệt độ trung bình hàng năm đang tăng dần theo thời gian và đặc biệt là tăng nhanh trong những năm gần đây. Ngoài ra, cũng thấy rằng nhiệt độ trung bình tháng có xu hướng giảm từ tháng 6 đến tháng 9 (mùa mưa), trong khi tăng từ tháng 12 đến tháng 2 (mùa đông).Cũng đã tìm thấy sự khác biệt rõ ràng về nhiệt độ giữa các mùa trong năm.
* Để dự đoán nhiệt độ trong tương lai, đã sử dụng mô hình Decision Tree Regressor và dự đoán được nhiệt độ trung bình cho năm 2023. Kết quả cho thấy rằng nhiệt độ trung bình dự kiến trong năm 2023 sẽ tiếp tục tăng.

1. **Hạn chế**

* Dữ liệu còn hạn chế và chưa phong phú
* Chưa thực hiện dự đoán với mốc thời gian ngắn ví dụ như 1 ngày, hoặc 1 tuần.
* Chưa đánh giá toán diện được các phương pháp thử nghiệm

1. **Hướng phát triển**

* Trong tương lai, nghiên cứu về thời tiết có thể được mở rộng bằng cách sử dụng các bộ dữ liệu khác như dữ liệu về mưa, độ ẩm, áp suất không khí, gió, v.v. để phân tích các yếu tố ảnh hưởng đến thời tiết. Cũng có thể sử dụng các mô hình dự đoán khác như ARIMA, LSTM, v.v. để dự đoán nhiệt độ trong tương lai và so sánh kết quả với Decision Tree Regressor.
* Ngoài ra, để hiểu rõ hơn về tác động của biến đổi khí hậu, có thể so sánh các kết quả với các bộ dữ liệu khác ở các vùng khác trên thế giới, để xem liệu có bị ảnh hưởng nặng nề hơn so với các nơi khác hay không.

# TÀI LIỆU THAM KHẢO

1. Indian Meteorological Department, "Climate of India," 2020. [Online]. Available: <http://www.indiaweather.gov.in/climate/climate.html>.
2. Sharma, A., and Sharma, S., "A Review on Machine Learning Techniques for Weather Forecasting," in 2022 International Conference on Data Science and Computational Intelligence (DSCI), 2022, pp. 73-78.
3. Saha, S., and Sharma, S., "Exploring the Relationships between Indian Weather and Agricultural Yield using Machine Learning," in 2022 IEEE International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), 2022, pp. 1-6.
4. Zhang, X., Guan, X., Zhang, L., and Chen, H., "A High-resolution Daily Temperature Dataset over India: Analysis of Variability, Trends and Extremes," in Journal of Climate, vol. 32, no. 11, pp. 3163-3183, 2019.
5. Santhosh, M., and Prasath, V. B. S., "Machine Learning Techniques for Indian Weather Analysis and Forecasting," in 2021 IEEE International Conference on Artificial Intelligence and Machine Learning in Engineering Applications (ICAIMLEA), 2021, pp. 154-158.
6. Pandey, A., and Pradhan, R. K., "Predicting Future Rainfall in India using Machine Learning Techniques," in 2022 IEEE 2nd International Conference on Inventive Research in Computing Applications (ICIRCA), 2022, pp. 1131-1135.