**TRƯỜNG ĐẠI HỌC SƯ PHẠM**

**ĐẠI HỌC ĐÀ NẴNG**

**KHOA TIN HỌC**



**BÁO CÁO MÔN HỌC**

**KHAI PHÁ DỮ LIỆU**

GIẢNG VIÊN HƯỚNG DẪN: NGUYỄN THỊ NGỌC ANH

SINH VIÊN THỰC HIỆN: PHAN QUANG THẮNG

**Đà Nẵng, 2022**

Mục Lục

[PHẦN I. LÝ THUYẾT 3](#_Toc105755529)

[**I.** **Khai phá dữ liệu** 3](#_Toc105755530)

[**1.** **Khái niệm khai phá dữ liệu** 3](#_Toc105755531)

[**2.** **Các lĩnh vực liên quan tới Khai phá dữ liệu** 3](#_Toc105755532)

[**3.** **Các ứng dụng thực tế của Khai phá dữ liệu:** 3](#_Toc105755533)

[**4.** **Khai phá dữ liệu với tìm kiếm thông thường** 3](#_Toc105755534)

[**II.** **II. Quy trình khai phá dữ liệu** 3](#_Toc105755535)

[**1.** **Tổng quan Quy trình Khai phá dữ liệu** 3](#_Toc105755536)

[**2.** **Nhiệm vụ của mỗi bước trong Quy trình khai phá dữ liệu** 3](#_Toc105755537)

[**3.** **Tiền xử lí dữ liệu:** 3](#_Toc105755538)

[**III.** **III. Kho dữ liệu** 3](#_Toc105755539)

[**1.** **Khái niệm** 3](#_Toc105755540)

[**2.** **Kho dữ liệu với Cơ sở dữ liệu tác nghiệp** 3](#_Toc105755541)

[**3.** **Đặc tính của Kho dữ liệu** 3](#_Toc105755542)

[**4.** **OLAP. OLTP và OLAP** 3](#_Toc105755543)

[**5.** **Phân biệt OLTP và OLAP** 3](#_Toc105755544)

[**6.** **Trình bày kiến trúc của Kho dữ liệu** 3](#_Toc105755545)

[**7.** **Trình bày quy trình xây dựng Kho dữ liệu** 3](#_Toc105755546)

[**Kết luận** 3](#_Toc105755547)

[**8.** **Mô hình hóa Kho dữ liệu.Sơ đồ ngôi sao, bông tuyết và chòm sao** 3](#_Toc105755548)

[**IV.** **IV. Học máy** 3](#_Toc105755549)

[**1.** **Khái niệm** 3](#_Toc105755550)

[**2.** **Các cơ chế học** 3](#_Toc105755551)

[**3.** **Sự liên hệ của Học máy với Khai phá dữ liệu** 3](#_Toc105755552)

[**4.** **Ứng dụng của Học máy** 3](#_Toc105755553)

[**V.** **V. Phân lớp (Classification)** 3](#_Toc105755554)

[**1.** **Khái niệm** 3](#_Toc105755555)

[**2.** **Mục đích của Phân lớp** 3](#_Toc105755556)

[**3.** **Các thuật ngữ liên quan tới bài toán Phân lớp** 3](#_Toc105755557)

[**4.** **Quy trình phân lớp: Vẽ mô hình và trình bày các bước** 3](#_Toc105755558)

[**5.** **Ví dụ** 3](#_Toc105755559)

[**VI.** **VI. Thuật toán K-NN** 3](#_Toc105755560)

[**1.** **Khái niệm** 3](#_Toc105755561)

[**2.** **Quy trình thực hiện Phân lớp dựa trên thuật toán KNN** 3](#_Toc105755562)

[**3.** **Viết mã giả** 3](#_Toc105755563)

[**4.** **Ví dụ minh họa:** 3](#_Toc105755564)

[**5.** **Viết chương trình (source code) cho bài toán được lấy minh họa ở phần 4.** 3](#_Toc105755565)

[**6.** **Trình bày Ưu/Nhược điểm của thuật toán KNN** 3](#_Toc105755566)

[**7.** **Ứng dụng thực tế: Nghiên cứu ứng dụng thuật toán KNN trong việc chẩn đoán bệnh tim của bệnh nhân. Thực hiện theo yêu cầu như sau:** 3](#_Toc105755567)

[**VII.** **Thuật toán Naïve Bayes:** 3](#_Toc105755568)

[**1.** **Khái niệm** 3](#_Toc105755569)

[**2.** **Viết mã giả** 3](#_Toc105755570)

[**3.** **Ví dụ minh họa:** 3](#_Toc105755571)

[**4.** **Chương trình (source code)** 3](#_Toc105755572)

[**5.** **Ưu,Nhược điểm của thuật toán Bayes** 3](#_Toc105755573)

[**VIII.** **Thuật toán Decision tree:** 3](#_Toc105755574)

[**1.** **Khái niệm/ Ý tưởng** 3](#_Toc105755575)

[**2.** **Quy trình thực hiện** 3](#_Toc105755576)

[**3.** **Viết mã giả** 3](#_Toc105755577)

[**4.** **Ví dụ minh họa** 3](#_Toc105755578)

[**5.** **Ưu/Nhược điểm của thuật toán Decision tree:** 3](#_Toc105755579)

[**IX.** **Phân cụm** 3](#_Toc105755580)

[**1.** **Khái niệm** 3](#_Toc105755581)

[**2.** **Mục đích của Phân cụm** 3](#_Toc105755582)

[**3.** **Phân cụm và Phân lớp** 3](#_Toc105755583)

[**X.** **Thuật toán K-means:** 3](#_Toc105755584)

[**1.** **Khái niệm/ Ý tưởng** 3](#_Toc105755585)

[**2.** **Quy trình thực hiện Phân lớp dựa trên thuật toán K-means** 3](#_Toc105755586)

[**3.** **Viết mã giả** 3](#_Toc105755587)

[**4.** **Ví dụ minh họa** 3](#_Toc105755588)

[**5.** **Viết chương trình (source code) cho bài toán được lấy minh họa ở phần 4.** 3](#_Toc105755589)

[**6.** **Ưu/Nhược điểm của thuật toán K-means** 3](#_Toc105755590)

[**XI.** **Luật kết hợp** 3](#_Toc105755591)

[**1.** **Khái niệm** 3](#_Toc105755592)

[**2.** **Các miền ứng dụng** 3](#_Toc105755593)

[**3.** **Thuật toán APRIORI:** 3](#_Toc105755594)

[**4.** **Thuật toán FP-GROWTH:** 3](#_Toc105755595)

[**XII.** **XII. Đánh giá mô hình** 3](#_Toc105755596)

[**1.** **Các phương thức để xác định hiệu năng của mộ hình Phân lớp/Phân cụm** 3](#_Toc105755597)

[**2.** **Phương pháp Confusion matrix** 3](#_Toc105755598)

[**3.** **Cách tính độ chính xác mô hình thông qua confusion matrix** 3](#_Toc105755599)

**Danh Mục Hình Ảnh**

[Hình 1. 1 Quy trình khai phá dữ liệu 7](#_Toc105857799)

[Hình 1. 2 Lược đồ hình sao của kho dữ liệu bán hàng 13](#_Toc105857800)

[Hình 1. 3 Lược đồ bông tuyết của kho dữ liệu bán hàng 13](#_Toc105857801)

[Hình 1. 4 Lược đồ chòm sao dữ kiện của kho dữ liệu bán hàng và vận chuyển 14](#_Toc105857802)

[Hình 1. 5 Quy trình phân lớp 17](#_Toc105857803)

[Hình 1. 6 chương trình (source code) cho bài toán được lấy minh họa ở phần 4. 20](#_Toc105857804)

[Hình 1. 7 Source Code Knn 21](#_Toc105857805)

[Hình 1. 8 Source Code Knn 21](#_Toc105857806)

[Hình 1. 9 Source Code Bayes 24](#_Toc105857807)

[Hình 1. 10 Source Code K-means 30](#_Toc105857808)

[Hình 2. 1Trực quan 2 thuộc tính 1 và 2 35](#_Toc105857809)

[Hình 2. 2Trực quan 2 thuộc tính 1 và 3 36](#_Toc105857810)

[Hình 2. 3Trực quan 2 thuộc tính 1 và 4 36](#_Toc105857811)

[Hình 2. 4 Trực quan 2 thuộc tính 2 và 3 37](#_Toc105857812)

[Hình 2. 5 Trực quan 2 thuộc tính 2 và 4 37](#_Toc105857813)

[Hình 2. 6 Trực quan 2 thuộc tính 3 và 4 38](#_Toc105857814)

[Hình 2. 7 Source Code Knn cho tập dữ liệu Iris 38](#_Toc105857815)

[Hình 2. 8 Source Code Bayes 39](#_Toc105857816)

[Hình 2. 9 Source Code K-means 44](#_Toc105857817)

# PHẦN I. LÝ THUYẾT

1. **Khai phá dữ liệu**
2. **Khái niệm khai phá dữ liệu**

Data mining (Khai phá dữ liệu): là quá trình khám phá ra những mối tương quan mới có ý nghĩa , mẫu và xu hướng bằng cách chọn ra từ lượng lớn dữ liệu đã được lưu trữ từ trước, sử dụng các công nghệ nhận diện mẫu,xác suất thống kê và các kĩ thuật toán học

1. **Các lĩnh vực liên quan tới Khai phá dữ liệu**

* Algorithms (thuật toán): Hỗ trợ khai phá tri thức từ cơ sở dữ liệu.
* Machine Learning (Học Máy): Là phương tiện để tiến hành khai phá dữ liệu. Sử dụng các dữ liệu từ DM để xây dựng các mô hình
* Database Management (Quản trị cơ sở dữ liệu): Lưu trữ dữ liệu để khai phá
* Artificial Intelligence (Trí tuệ nhân tạo): Khai phá dữ liệu là một phần của mã lập trình với thông tin và dữ liệu cần thiết cho các hệ thống AI
* Visualization (Trực Quan dữ liệu) : Mô tả dữ liệu một cách trực quan nhất

1. **Các ứng dụng thực tế của Khai phá dữ liệu:**

Lĩnh vực giao thông vận tải

Cho phép tìm ra các xe lưu thông ngược chiều , vượt đèn đỏ,… thông qua xử lí dữ liệu hình ảnh

Lĩnh vực thương mại điện tử

Nhiều công ty thương mại điện tử đang áp dụng ứng dụng của Data Mining để bán hàng qua nhiều nước thông qua các trang web của họ. Một trong những công ty nổi tiếng nhất ứng dụng điều này là Amazon. Họ sử dụng các kỹ thuật khai phá dữ liệu để lái “những người đã xem sản phẩm đó cũng thích sản phẩm được giới thiệu này”.

Lĩnh vực chăm sóc sức khỏe

Data Mining giúp các doanh nghiệp gia tăng lợi thế cạnh tranh bằng việc cung cấp các thông tin về chính phủ, quy định và đối thủ cạnh tranh. Nó hỗ trợ quá trình R&D và sau đó là chiến lược tiếp cận thị trường với khả năng tiếp cận thông tin nhanh chóng ở mọi giai đoạn.

Thêm vào đó, việc tìm ra mối quan hệ giữa các loại bệnh và hiệu quả của phương pháp điều trị sẽ giúp thay đổi các loại thuốc mới hoặc đảm bảo rằng bệnh nhân được chăm sóc phù hợp, kịp thời.

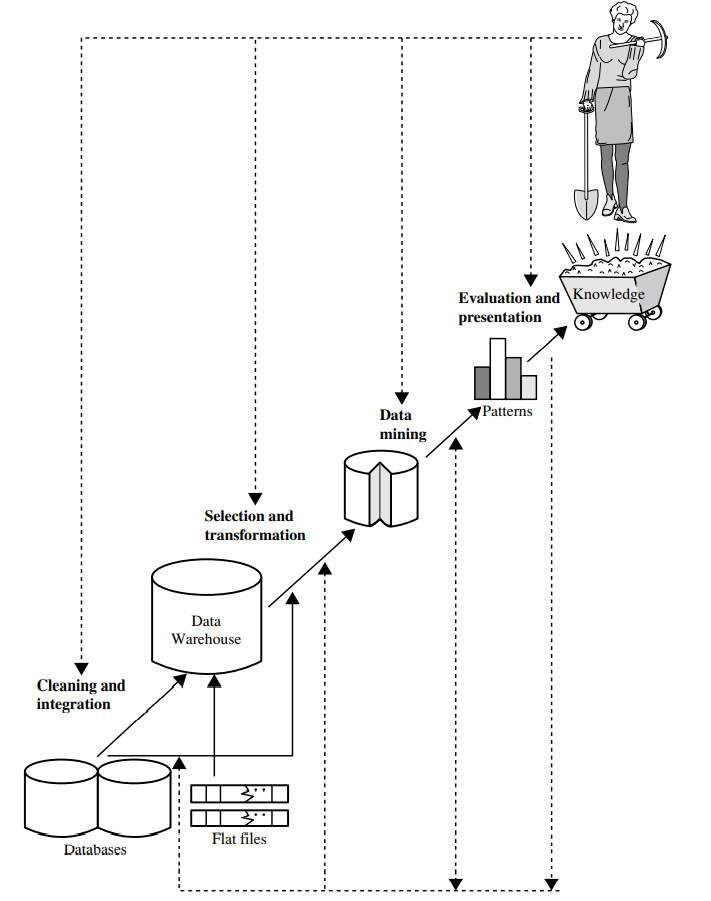
1. **Khai phá dữ liệu với tìm kiếm thông thường**

Khai phá dữ liệu là tìm ra mối tương quan mới , xu hướng,.. từ lượng lớn dữ liệu lưu trữ từ trước thông qua các phương pháp nhận diện mẫu, xác xuất thông kê và các kĩ thuật toán học còn tìm kiếm thông thường chỉ đơn giản lọc ra dữ liệu từ dữ liệu có sẵn.

Ví dụ: Ví dụ về việc tìm kiếm biển số xe vi phạm luật giao thông. Đối với tìm kiếm thông thường bạn chỉ có thể tìm ra bằng cách biết trước biển số xe vi phạm. Còn khai phá dữ liệu có thể thông qua hình ảnh trích xuất từ camera sau đó xử lí và cho ra xe lưu thông có vi phạm hay không.

## **Quy trình khai phá dữ liệu**

1. **Tổng quan Quy trình Khai phá dữ liệu**



Hình 1. 1 Quy trình khai phá dữ liệu

1. **Nhiệm vụ của mỗi bước trong Quy trình khai phá dữ liệu**

Cleaning data(Làm sạch dữ liệu): để loại bỏ nhiễu và dữ liệu không nhất quán.

Data integration (Tích hợp dữ liệu): trong đó nhiều nguồn dữ liệu có thể được kết hợp

Data selection (Lựa chọn dữ liệu) :trong đó dữ liệu liên quan đến nhiệm vụ phân tích được truy xuất từ cơ sở dữ liệu.

Data transformation (Chuyển đổi dữ liệu):trong đó dữ liệu được chuyển đổi và hợp nhất thành các biểu mẫu thích hợp để khai thác bằng cách thực hiện các hoạt động tóm tắt hoặc tổng hợp.

Data Mining (Khai thác dữ liệu) :một quy trình thiết yếu trong đó các phương pháp thông minh được áp dụng để trích xuất mẫu dữ liệu.

Patternevaluation (Đánh giá mẫu) :để xác định độ thú vị của mẫu dựa theo các thước đo.

Knowledge presentation (trình bày kiến thức): nơi các kỹ thuật công nghệ trực quan và trình bày kiến thức được sử dụng để trình bày kiến thức đã khai thác cho người dùng.

1. **Tiền xử lí dữ liệu:**

Các nhiệm vụ chính trong tiền xử lý dữ liệu

- Data cleaning

+ Điền vào các giá trị còn thiếu, làm mịn dữ liệu nhiễu, xác định hoặc loại bỏ các ngoại lệ và giải quyết các mâu thuẫn - Tích hợp dữ liệu

+ Tích hợp nhiều cơ sở dữ liệu, khối dữ liệu hoặc tệp - Chuyển đổi dữ liệu

+ Normalization and aggregation — Data reduction:Thu được giảm đại diện về khối lượng nhưng tạo ra các kết quả phân tích giống nhau hoặc tương tự - Tùy theo dữ liệu.

+ Một phần của việc giảm dữ liệu nhưng có tầm quan trọng đặc biệt,đặc biệt là đối với dữ liệu số

-Feature reduction:

+Là quá trình giảm số chiều của dữ liệu mà không làm mất các thông tin quan trọng.

-Feature selection:

+Là quá trình lựa chọn một tập hợp con các đặc điểm có liên quan để sử dụng trong việc xây dựng mô hình

VD:

## **Kho dữ liệu**

### **Khái niệm**

Một kho dữ liệu là kho thông tin được thu thập từ nhiều nguồn, được lưu trữ dưới một thể thống nhất

lược đồ và thường nằm ở một trang web. Kho dữ liệu được xây dựng thông qua

quy trình làm sạch dữ liệu, tích hợp dữ liệu, chuyển đổi dữ liệu, tải dữ liệu và làm mới dữ liệu chuẩn bị

### **Kho dữ liệu với Cơ sở dữ liệu tác nghiệp**

Nhiệm vụ chính yếu của các hệ thống cơ sở dữ liệu tác nghiệp trực tuyến là thực hiện các giao dịch trực tuyến và xử lý truy vấn. Các hệ thống này được gọi là các hệ thống xử lý giao dịch trực tuyến (On-line transaction processing system), được viết tắt là OLTP. Các hệ thống này tập trung vào các hoạt động hằng ngày của một tổ chức như là mua bán, kiểm kê, sản xuất, công việc ngân hàng, bảng lương, đăng kí, kế toán. Trong khi đó, các kho dữ liệu có nhiệm vụ là phân tích dữ liệu và hỗ trợ quyết định. Các hệ thống như vậy được tổ chức và biểu diễn dữ liệu dưới nhiều định dạng khác nhau để có thể đáp ứng các nhu cầu khác nhau của người dùng. Những hệ thống này đươc gọi là các hệ thống xử lý phân tích trực tuyến (On-line analytical processing system), viết tắt là OLAP. Các đặc điểm giúp phân biệt giữa OLAP và OLTP là kích thước cơ sở dữ liệu, tính thường xuyên của thao tác, độ đo thực thi.

### **Đặc tính của Kho dữ liệu**

Hướng theo chủ đề: Kho dữ liệu được tổ chức xung quanh các chủ thể chính như khách hàng, nhà cung cấp, sản phẩm và bán hàng. Thay vì tập trung vào ngày này qua ngày khác hoạt động và xử lý giao dịch của một tổ chức, tập trung vào kho dữ liệu về mô hình hóa và phân tích dữ liệu cho những người ra quyết định. Do đó, kho dữ liệu thường cung cấp một cái nhìn đơn giản và ngắn gọn về các vấn đề chủ đề cụ thể bằng cách loại trừ dữ liệu không hữu ích trong quá trình hỗ trợ quyết định.

Tích hợp: Kho dữ liệu thường được xây dựng bằng cách tích hợp nhiều nguồn không đồng nhất, chẳng hạn như cơ sở dữ liệu quan hệ, tệp và hồ sơ giao dịch trực tuyến Các kỹ thuật làm sạch dữ liệu và tích hợp dữ liệu được áp dụng để đảm bảo tính nhất quán trong các quy ước đặt tên, cấu trúc mã hóa, các biện pháp thuộc tính, v.v.

Biến thể theo thời gian: Dữ liệu được lưu trữ để cung cấp thông tin từ góc độ lịch sử (ví dụ: 5–10 năm qua). Mọi cấu trúc chính trong kho dữ liệu đều chứa ngầm định hoặc rõ ràng, một yếu tố thời gian.

Không linh hoạt: Kho dữ liệu luôn là một kho lưu trữ dữ liệu riêng biệt về mặt vật lý được chuyển đổi từ dữ liệu ứng dụng được tìm thấy trong môi trường hoạt động. Bởi vì sự tách biệt này, một kho dữ liệu không yêu cầu xử lý giao dịch, khôi phục, và các cơ chế kiểm soát đồng thời. Nó thường chỉ yêu cầu hai thao tác trong truy cập dữ liệu : tải dữ liệu ban đầu và truy cập dữ liệu.

### **OLAP. OLTP và OLAP**

Nhiệm vụ chính của các hệ thống cơ sở dữ liệu hoạt động trực tuyến là thực hiện các giao dịch và xử lý truy vấn trực tuyến. Các hệ thống này được gọi là xử lý giao dịch trực tuyến (OLTP- online transaction processing) hệ thống. Chúng bao gồm hầu hết các hoạt động hàng ngày của một tổ chức như như mua hàng, hàng tồn kho, sản xuất, ngân hàng, bảng lương, đăng ký và kế toán. Mặt khác, hệ thống kho dữ liệu phục vụ người dùng hoặc nhân viên tri thức trong vai trò của phân tích dữ liệu và ra quyết định. Hệ thống như vậy có thể tổ chức và trình bày dữ liệu ở nhiều định dạng khác nhau để đáp ứng nhu cầu đa dạng của những người dùng khác nhau. Các hệ thống này được gọi là hệ thống xử lý phân tích trực tuyến (OLAP- online analytical processing).

### **Phân biệt OLTP và OLAP**

Người dùng và định hướng hệ thống: Hệ thống OLTP hướng tới khách hàng và được sử dụng để xử lý giao dịch và truy vấn của nhân viên, khách hàng và công nghệ thông tin các chuyên gia. Hệ thống OLAP hướng tới thị trường và được sử dụng để phân tích dữ liệu bằng cách nhân viên tri thức, bao gồm các nhà quản lý, giám đốc điều hành và nhà phân tích.

Nội dung dữ liệu: Hệ thống OLTP quản lý dữ liệu hiện tại, thường là quá chi tiết để dễ dàng sử dụng cho việc ra quyết định. Hệ thống OLAP quản lý một lượng lớn dữ liệu lịch sử, cung cấp các phương tiện để tóm tắt và tổng hợp, đồng thời lưu trữ và quản lý thông tin ở các mức độ chi tiết khác nhau. Các tính năng này làm cho dữ liệu dễ sử dụng hơn để đưa ra quyết định sáng suốt.

Thiết kế cơ sở dữ liệu: Hệ thống OLTP thường sử dụng dữ liệu mối quan hệ thực thể (ER) mô hình và thiết kế cơ sở dữ liệu hướng ứng dụng. Một hệ thống OLAP thường áp dụng mô hình ngôi sao hoặc bông tuyết và hướng chủ thể thiết kế cơ sở dữ liệu.

Chế độ xem: Hệ thống OLTP chủ yếu tập trung vào dữ liệu hiện tại trong một doanh nghiệp hoặc bộ phận, mà không đề cập đến dữ liệu lịch sử hoặc dữ liệu trong các tổ chức khác nhau. Trong ngược lại, một hệ thống OLAP thường kéo dài nhiều phiên bản của một lược đồ cơ sở dữ liệu, do quá trình phát triển của một tổ chức. Hệ thống OLAP cũng xử lý thông tin có nguồn gốc từ các tổ chức khác nhau, tích hợp thông tin từ nhiều kho dữ liệu. Do khối lượng lớn của chúng, dữ liệu OLAP được lưu trữ trên nhiều bộ nhớ phương tiện truyền thông.

Các mẫu truy cập: Các mẫu truy cập của hệ thống OLTP chủ yếu bao gồm ngắn gọn, giao dịch nguyên tử. Một hệ thống như vậy yêu cầu cơ chế kiểm soát và phục hồi đồng thời. Tuy nhiên, quyền truy cập vào hệ thống OLAP chủ yếu là các hoạt động chỉ đọc (bởi vì hầu hết các kho dữ liệu lưu trữ thông tin lịch sử thay vì cập nhật), mặc dù nhiều có thể là các truy vấn phức tạp.

### **Trình bày kiến trúc của Kho dữ liệu**

Kiến trúc kho dữ liệu rất phức tạp vì nó là một hệ thống thông tin chứa dữ liệu lịch sử và dữ liệu giao hoán từ nhiều nguồn. Có 3 cách tiếp cận để xây dựng các lớp Kho dữ liệu: Một lớp, Hai lớp và Ba lớp. Kiến trúc 3 tầng này của Data Warehouse được giải thích như bên dưới.

* Kiến trúc một tầng

Mục tiêu của một lớp là giảm thiểu lượng dữ liệu được lưu trữ. Mục tiêu này là để loại bỏ dư thừa dữ liệu. Kiến trúc này không được sử dụng thường xuyên trong thực tế.

* Kiến trúc hai tầng

Kiến trúc hai lớp là một trong những lớp Kho dữ liệu phân tách các nguồn vật lý có sẵn và kho dữ liệu. Kiến trúc này không thể mở rộng và cũng không hỗ trợ một số lượng lớn người dùng cuối. Nó cũng có vấn đề kết nối vì giới hạn mạng.

* Kiến trúc kho dữ liệu ba tầng

Đây là Kiến trúc Kho dữ liệu được sử dụng rộng rãi nhất.

Nó bao gồm Tầng trên cùng, Tầng giữa và Tầng dưới cùng.

Tầng đáy: Cơ sở dữ liệu của các máy chủ Datawarehouse ở tầng dưới cùng. Nó thường là một hệ thống cơ sở dữ liệu quan hệ. Dữ liệu được làm sạch, chuyển đổi và tải vào lớp này bằng các công cụ back-end.

Tầng giữa: Tầng giữa trong Kho dữ liệu là một máy chủ OLAP được triển khai bằng cách sử dụng mô hình ROLAP hoặc MOLAP. Đối với người dùng, tầng ứng dụng này trình bày một dạng xem trừu tượng của cơ sở dữ liệu. Lớp này cũng hoạt động như một trung gian giữa người dùng cuối và cơ sở dữ liệu.

Tầng trên: Tầng trên cùng là lớp khách hàng giao diện người dùng. Cấp cao nhất là các công cụ và API mà bạn kết nối và lấy dữ liệu ra từ kho dữ liệu. Nó có thể là Công cụ truy vấn, công cụ báo cáo, công cụ truy vấn được quản lý, công cụ Phân tích và công cụ Khai thác dữ liệu.

### **Trình bày quy trình xây dựng Kho dữ liệu**

* Thu thập yêu cầu và thiết kế kho dữ liệu

Thiết kế một kho dữ liệu là một hành trình rộng lớn của doanh nghiệp. Kho dữ liệu liên quan đến tất cả các lĩnh vực của doanh nghiệp, vì vậy mọi bộ phận cần phải phù hợp với cấu trúc và thiết kế.

Ví dụ: nếu dữ liệu tiếp thị và bán hàng của bạn không được kết hợp và liên kết, truy vấn tổng thể của bạn sẽ bỏ sót các thành phần quan trọng.

Các yêu cầu và thiết kế kho dữ liệu nên tập trung vào các mục tiêu cốt lõi sau:

Điều chỉnh các mục tiêu của dự án với các mục tiêu của bộ phận

Xác định phạm vi của dự án trong mối quan hệ với mục tiêu của doanh nghiệp

Hiểu trạng thái hiện tại và ngăn xếp công nghệ trong tương lai của bạn

Suy nghĩ về các lớp bảo mật bắt buộc

Giảm thiểu rủi ro

* Thiết lập môi trường vật chất doanh nghiệp

Khi các yêu cầu từ kho dữ liệu đã rõ ràng, điều cần thiết là phải thiết lập cơ sở dữ liệu và máy chủ. Ở mức tối thiểu, điều quan trọng là phải thiết lập một môi trường phát triển và một môi trường sản xuất. Trong hầu hết các dự án kho dữ liệu, có ba môi trường: Phát triển, Kiểm tra và Sản xuất. Việc thiết lập các môi trường khác nhau là rất quan trọng vì những lý do sau:

Tất cả các thay đổi có thể được kiểm tra và đảm bảo chất lượng trước tiên mà không ảnh hưởng đến môi trường sản xuất

Đảm bảo chất lượng và phát triển có thể xảy ra trong thời gian người dùng truy cập vào kho dữ liệu

Khi tồn tại bất kỳ câu hỏi hoặc truy vấn nào xung quanh dữ liệu, việc có các môi trường riêng biệt cho phép nhóm kho dữ liệu kiểm tra dữ liệu mà không có tác động trực tiếp đến môi trường sản xuất.

* Mô hình hóa dữ liệu và cách nó liên quan đến việc xây dựng kho dữ liệu

Mô hình hóa kho dữ liệu là quá trình cấu trúc và thiết kế các lược đồ của thông tin chi tiết cho kho dữ liệu.

Mục tiêu của mô hình hóa kho dữ liệu là phát triển các lược đồ mô tả thực tế theo khả năng tốt nhất của hệ thống, mà kho dữ liệu cần thiết để tạo điều kiện và hỗ trợ.

Có hai lý do chính giải thích tại sao mô hình kho dữ liệu là một thành phần thực sự quan trọng và là một phần của việc xây dựng kho dữ liệu vào năm 2021 và hơn thế nữa.

Các khách hàng của kho dữ liệu, lược đồ có thể trực quan hóa các mối quan hệ giữa các dữ liệu kho để sử dụng chúng dễ dàng hơn.

Lược đồ được thiết kế tốt cho phép xuất hiện cấu trúc kho dữ liệu hiệu quả, nâng cao hiệu quả của dữ liệu và giúp giảm chi phí triển khai kho.

* Chọn Giải pháp Trích xuất, Chuyển và Nạp (ETL) của bạn một cách cẩn thận

Quá trình trích xuất và tổ chức dữ liệu thô, biến đổi nó để làm cho nó dễ hiểu hơn nhiều và tải nó vào cơ sở dữ liệu được gọi là quá trình ETL.

Dữ liệu đến từ các nguồn dữ liệu khác nhau có cấu trúc riêng biệt, mọi tập dữ liệu cần được chuyển đổi riêng biệt trước khi sử dụng để phân tích.

Vì quy trình ETL chịu trách nhiệm chính cho công việc ở giữa, việc chọn một tiêu chí phụ hoặc phát triển một quy trình ETL kém có thể phá vỡ toàn bộ kho hàng của bạn.

Nếu bạn là một CXO đưa ra các quyết định, bạn chủ yếu nên nhắm mục tiêu trực quan tốt, tốc độ tối ưu và khả năng xây dựng đường ống dẫn dữ liệu nhất quán giữa giải pháp kho mới của chúng tôi và kiến trúc hiện có của bạn.

* Khối lập phương OLAP

Xử lý phân tích trực tuyến là một danh mục phần mềm cho phép người dùng phân tích thông tin từ nhiều DB cùng một lúc. Nó tạo điều kiện thuận lợi cho việc xem dữ liệu kinh doanh từ các quan điểm khác nhau và là một công cụ công nghệ quan trọng cho các nhà phân tích. Hầu hết các nhà phân tích kinh doanh cần thường xuyên nhóm, tham gia và tổng hợp dữ liệu. Với dữ liệu OLAP có thể được tổng hợp trước và tính toán trước, giúp phân tích nhanh hơn. Cơ sở dữ liệu OLAP được chia thành một hoặc nhiều khối. OLAP DB’s được chia thành nhiều khối để cho phép phân tích dữ liệu nhanh hơn theo nhiều thứ nguyên xác định một doanh nghiệp và các vấn đề của nó. Ưu điểm của Hình khối OLAP là nó có thể được coi như một phần mở rộng của cấu trúc mô hình được cung cấp bởi bảng tính, chứa dữ liệu trong các cột và hàng, tức là mảng dữ liệu hai chiều. Phần tốt nhất là Khối lập phương OLAP có thể chứa bất kỳ số lượng mảng hoặc Thứ nguyên nào.

* Tạo Giao diện người dùng của Kho dữ liệu

Bất kể sức mạnh của OLAP như thế nào, nếu bạn không thể dễ dàng hình dung các báo cáo, kho dữ liệu sẽ không mang lại giá trị gì cho doanh nghiệp. Do đó, là một phần của sáng kiến lưu trữ dữ liệu, điều cần thiết là phải tập trung vào phát triển giao diện người dùng.

Việc phát triển giao diện người dùng nằm trong một lớp vỏ cách người dùng truy cập dữ liệu để phân tích và chạy các báo cáo tương ứng.

Có nhiều lựa chọn có sẵn, bao gồm xây dựng giao diện người dùng của bạn thông qua các chuyên gia nội bộ hoặc mua nó từ kệ. Các động lực chính dẫn đến quyết định nên bao gồm:

Một trong những cân nhắc chính của giao diện người dùng nên bao gồm quyền truy cập từ bất kỳ thiết bị nào - máy tính bảng, máy tính để bàn, máy tính xách tay và điện thoại di động.

Công cụ này sẽ cho phép nhóm phát triển của bạn sửa đổi cấu trúc phụ trợ khi các yêu cầu báo cáo cấp doanh nghiệp thay đổi.

Giao diện người dùng đồ họa (GUI) cho phép người dùng dễ dàng tùy chỉnh báo cáo của họ khi cần.

* Thiết lập một bản giới thiệu

Sau khi bạn đã sẵn sàng và đã thực hiện đảm bảo chất lượng nghiêm ngặt kho dữ liệu, đã đến lúc giáo dục và đào tạo các bên liên quan trong doanh nghiệp của bạn, những người sẽ đưa ra các quyết định kinh doanh quan trọng dựa trên các điểm dữ liệu này.

### **Kết luận**

Đây là lộ trình chung mà hầu hết các doanh nghiệp sử dụng khi xây dựng kho dữ liệu.Bạn có thể đi sâu tìm hiểu các yêu cầu cụ thể đối với quy trình phát triển kho dữ liệu.

1. **Mô hình hóa Kho dữ liệu.Sơ đồ ngôi sao, bông tuyết và chòm sao**

Mô hình hóa kho dữ liệu là quá trình thiết kế các lược đồ thông tin chi tiết và tóm tắt của kho dữ liệu. Mục tiêu của mô hình hóa kho dữ liệu là phát triển một lược đồ mô tả thực tế, hoặc ít nhất là một phần của thực tế, mà kho dữ liệu cần hỗ trợ.

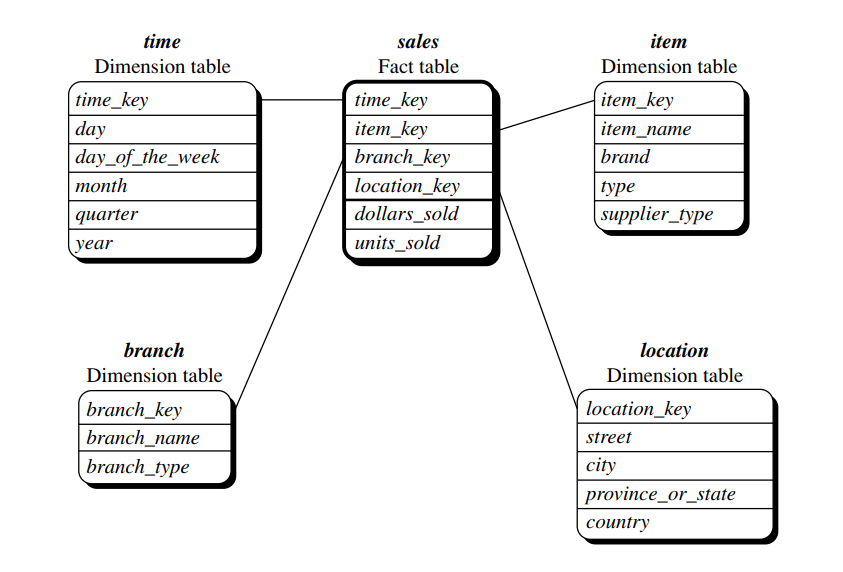
Mô hình hóa kho dữ liệu là một giai đoạn thiết yếu của việc xây dựng một kho dữ liệu vì hai lý do chính. Thứ nhất, thông qua lược đồ, khách hàng của kho dữ liệu có thể hình dung mối quan hệ giữa các dữ liệu kho để sử dụng chúng một cách dễ dàng hơn. Thứ hai, một lược đồ được thiết kế tốt cho phép xuất hiện cấu trúc kho dữ liệu hiệu quả, giúp giảm chi phí triển khai kho và nâng cao hiệu quả sử dụng.

Mô hình hóa dữ liệu trong kho dữ liệu khác với mô hình hóa dữ liệu trong các hệ thống cơ sở dữ liệu hoạt động. Chức năng chính của kho dữ liệu là hỗ trợ các quy trình DSS. Do đó, mục tiêu của mô hình kho dữ liệu là làm cho kho dữ liệu hỗ trợ hiệu quả các truy vấn phức tạp về thông tin dài hạn.

Ngược lại, mô hình hóa dữ liệu trong các hệ thống cơ sở dữ liệu hoạt động nhắm mục tiêu hỗ trợ hiệu quả các giao dịch đơn giản trong cơ sở dữ liệu như truy xuất, chèn, xóa và thay đổi dữ liệu. Hơn nữa, kho dữ liệu được thiết kế cho khách hàng có kiến ​​thức thông tin chung về doanh nghiệp, trong khi hệ thống cơ sở dữ liệu hoạt động được các chuyên gia phần mềm hướng tới sử dụng nhiều hơn để tạo ra các ứng dụng riêng biệt.

1. **Lược đồ hình sao**

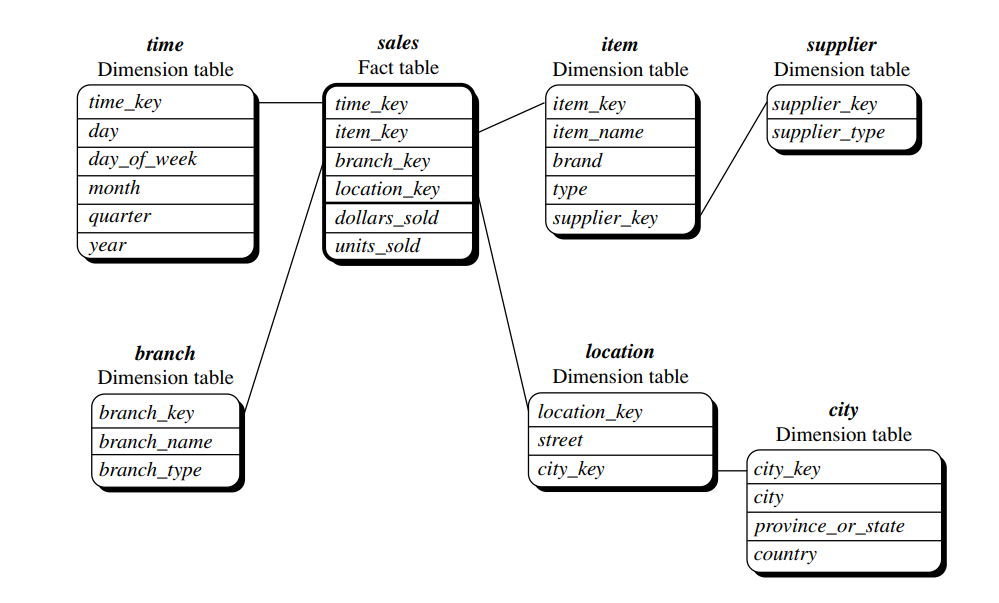
Mô hình mô hình hóa phổ biến nhất là lược đồ hình sao, trong đó kho dữ liệu chứa một bảng trung tâm lớn (bảng dữ kiện) chứa phần lớn dữ liệu, không dư thừa và một tập hợp các bảng phụ nhỏ hơn (thứ nguyên bảng), một bảng cho mỗi thứ nguyên. Biểu đồ giản đồ giống như một ngôi sao, với bảng kích thước được hiển thị theo mô hình xuyên tâm xung quanh bảng dữ kiện trung tâm.



Hình 1. 2 Lược đồ hình sao của kho dữ liệu bán hàng

1. **Lược đồ bông tuyết**

Lược đồ bông tuyết là một biến thể của mô hình giản đồ hình sao,trong đó một số bảng thứ nguyên được chuẩn hóa, do đó chia nhỏ dữ liệu thành bảng bổ sung. Biểu đồ lược đồ kết quả tạo thành một hình dạng tương tự như một bông tuyết

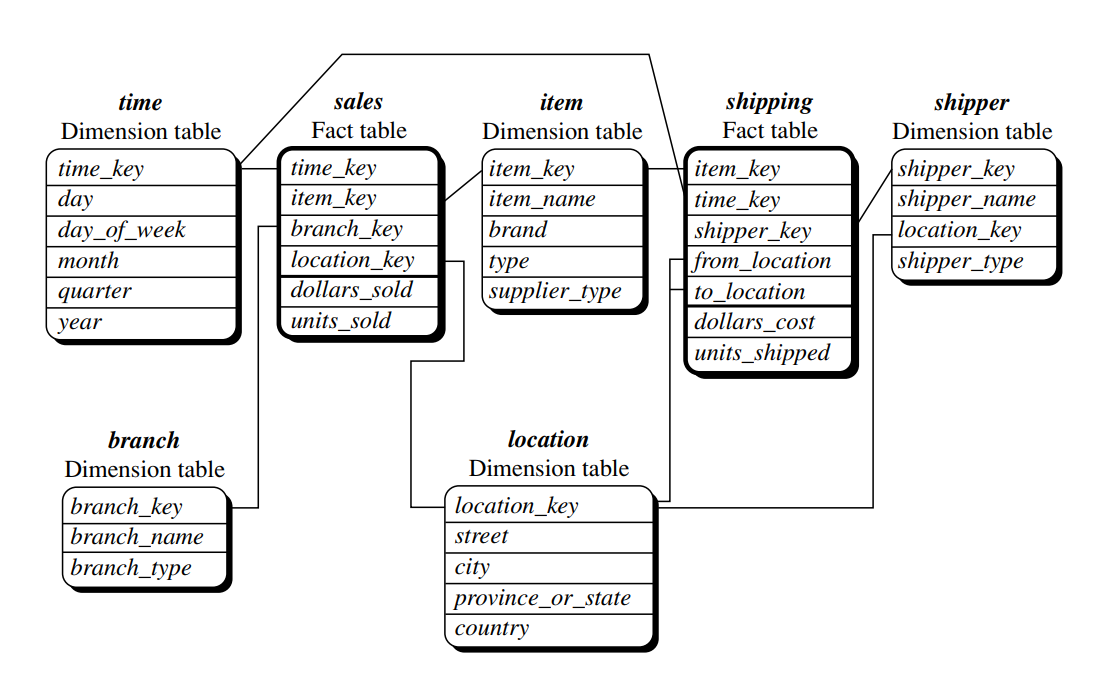


Hình 1. 3 Lược đồ bông tuyết của kho dữ liệu bán hàng

.

1. **Lược đồ chòm sao**

Các ứng dụng phức tạp có thể yêu cầu nhiều bảng dữ kiện để chia sẻ bảng kích thước. Loại lược đồ này có thể được xem như một tập hợp các ngôi sao và do đó được gọi là giản đồ thiên hà hoặc chòm sao thực tế.



Hình 1. 4 Lược đồ chòm sao dữ kiện của kho dữ liệu bán hàng và vận chuyển

## **Học máy**

### **Khái niệm**

Máy học nghiên cứu cách máy tính có thể học (hoặc cải thiện hiệu suất của chúng) dựa trên dữ liệu. Lĩnh vực nghiên cứu chính là dành cho các chương trình máy tính tự động học cách nhận ra các mẫu phức tạp và đưa ra quyết định thông minh dựa trên dữ liệu.

Ví dụ, một vấn đề học máy điển hình là lập trình máy tính để nó có thể tự động nhận dạng mã bưu điện viết tay trên thư sau khi học từ một tập hợp các ví dụ.

### **Các cơ chế học**

* Học tập có giám sát

Sự giám sát trong việc học đến từ các ví dụ được gắn nhãn trong tập dữ liệu đào tạo. Ví dụ, trong vấn đề nhận dạng mã bưu chính, một tập hợp các hình ảnh mã bưu chính viết tay và các bản dịch tương ứng có thể đọc được bằng máy của chúng được sử dụng làm ví dụ đào tạo, trong đó giám sát việc học tập của mô hình phân loại.

* Học không giám sát

Về cơ bản khá giống với phân cụm. Quá trình học tập không được giám sát vì các ví dụ đầu vào không được gắn nhãn lớp. Thông thường, chúng ta có thể sử dụng phân cụm để khám phá các lớp trong dữ liệu. Ví dụ, học tập không có giám sát phương thức có thể lấy, làm đầu vào, một tập hợp các hình ảnh của các chữ số viết tay. Giả sử rằng nó tìm thấy 10 cụm dữ liệu. Các cụm này có thể tương ứng với 10 chữ số khác biệt của 0 đến 9, tương ứng.

* Học bán giám sát

Trong khoa học máy tính, học nửa giám sát là một lớp của kỹ thuật học máy, sử dụng cả dữ liệu đã gán nhãn và chưa gán nhãn để huấn luyện - điển hình là một lượng nhỏ dữ liệu có gán nhãn cùng với lượng lớn dữ liệu chưa gán nhãn. Học nửa giám sát đứng giữa học không giám sát và có giám sát. Nhiều nhà nghiên cứu nhận thấy dữ liệu không gán nhãn, khi được sử dụng kết hợp với một chút dữ liệu có gán nhãn, có thể cải thiện đáng kể độ chính xác. Để gán nhãn dữ liệu cho một bài toán học máy thường đòi hỏi một chuyên viên có kĩ năng để phân loại bằng tay các ví dụ huấn luyện. Chi phí cho quy trình này khiến tập dữ liệu được gán nhãn hoàn toàn trở nên không khả thi, trong khi dữ liệu không gán nhãn thường tương đối rẻ tiền. Trong tình huống đó, học nửa giám sát có giá trị thực tiễn lớn lao.

* Học tăng cường

Phương pháp học máy cho phép người dùng đóng một vai trò tích cực trong quá trình học tập. Phương pháp học tập tích cực có thể hỏi người dùng (ví dụ: miền chuyên gia) để gắn nhãn một ví dụ, có thể từ một tập hợp các ví dụ không được gắn nhãn hoặc tổng hợp bởi chương trình học. Mục tiêu là tối ưu hóa chất lượng mô hình bằng cách tích cực thu thập kiến thức từ người dùng, với một hạn chế về số lượng các ví dụ họ có thể được yêu cầu dán nhãn.

### **Sự liên hệ của Học máy với Khai phá dữ liệu**

Khai phá dữ liệu và Máy học là những lĩnh vực chịu ảnh hưởng của nhau, mặc dù chúng có nhiều điểm chung nhưng lại có những điểm chung khác nhau.

Khai thác dữ liệu được con người thực hiện trên một số tập dữ liệu nhất định để tìm ra các mẫu thú vị giữa các mục trong tập dữ liệu. Khai phá dữ liệu sử dụng các kỹ thuật được tạo ra bởi máy học để dự đoán kết quả trong khi máy học là khả năng của máy tính học từ một tập dữ liệu đã được khai phá.

Các thuật toán học máy lấy thông tin đại diện cho mối quan hệ giữa các mục trong tập dữ liệu và tạo ra các mô hình để dự đoán kết quả trong tương lai. Những mô hình này không gì khác hơn là các hành động sẽ được thực hiện bởi máy để đạt được kết quả.

### **Ứng dụng của Học máy**

* *Nhận dạng hình ảnh:*

Nhận dạng hình ảnh là một trong những ứng dụng phổ biến nhất của học máy. Nó được sử dụng để xác định đối tượng, người, địa điểm, hình ảnh kỹ thuật số, v.v. Trường hợp sử dụng phổ biến của nhận dạng hình ảnh và nhận diện khuôn mặt là, Gợi ý gắn thẻ bạn bè tự động :

Facebook cung cấp cho chúng ta tính năng gợi ý tự động gắn thẻ bạn bè. Bất cứ khi nào chúng tôi tải lên một bức ảnh với bạn bè trên Facebook, chúng tôi sẽ tự động nhận được đề xuất gắn thẻ với tên và công nghệ đằng sau điều này là thuật toán nhận dạng và phát hiện khuôn mặt của máy học .

Nó dựa trên dự án Facebook có tên " Deep Face ", chịu trách nhiệm nhận dạng khuôn mặt và nhận dạng người trong ảnh.

* *Nhận dạng giọng nói*

Trong khi sử dụng Google, chúng tôi nhận được tùy chọn " Tìm kiếm bằng giọng nói ", tùy chọn này có tính năng nhận dạng giọng nói và là một ứng dụng phổ biến của máy học.

Nhận dạng giọng nói là một quá trình chuyển đổi hướng dẫn bằng giọng nói thành văn bản và nó còn được gọi là " Lời nói thành văn bản " hoặc " Nhận dạng giọng nói của máy tính ". Hiện tại, các thuật toán học máy được sử dụng rộng rãi bởi các ứng dụng khác nhau của nhận dạng giọng nói. Trợ lý Google , Siri , Cortana và Alexa đang sử dụng công nghệ nhận dạng giọng nói để làm theo hướng dẫn bằng giọng nói.

* *Trợ lý ảo cá nhân:*

Chúng tôi có nhiều trợ lý ảo cá nhân khác nhau như trợ lý Google , Alexa , Cortana , Siri . Như tên cho thấy, chúng giúp chúng tôi tìm kiếm thông tin bằng cách sử dụng hướng dẫn bằng giọng nói của chúng tôi. Những trợ lý này có thể giúp chúng tôi theo nhiều cách khác nhau chỉ bằng hướng dẫn bằng giọng nói của chúng tôi, chẳng hạn như Phát nhạc, gọi cho ai đó, Mở email, Lên lịch cuộc hẹn, v.v.

Các trợ lý ảo này sử dụng các thuật toán học máy như một phần quan trọng.

Các trợ lý này ghi lại các hướng dẫn bằng giọng nói của chúng tôi, gửi nó qua máy chủ trên một đám mây và giải mã nó bằng các thuật toán ML và hành động theo đó.

## **Phân lớp (Classification)**

### **Khái niệm**

Phân lớp là quá trình tìm kiếm một mô hình (hoặc chức năng) mô tả và phân loại các lớp hoặc khái niệm dữ liệu. Mô hình được suy ra dựa trên việc phân tích tập dữ liệu huấn luyện (các mẫu đã biết nhãn). Mô hình sau đó được sử dụng để dự đoán nhãn(hoặc lớp) của các mẫu chưa có nhãn.

### **Mục đích của Phân lớp**

Qua phân loại, loài người tổ chức được các sự vật, vật chất, hiện tượng, con người, động vật, thực vật, … thành các lớp. Lớp chính là một tập hợp các đơn vị, thành tố có chung một, hoặc một số đặc điểm, ví dụ sắt, đồng chì, kẽm có chung đặc đặc điểm là kim loại. Cơ sở để chia lớp là những đặc tính giống nhau của sự vật và hiện tượng. Dựa vào phương pháp đó, người ta có thể phân chia tiếp theo thành các lớp con hoặc phân lớp khác nhau theo một trật tự nhất định.

### **Các thuật ngữ liên quan tới bài toán Phân lớp**

Instance/Sample: mẫu, là dữ liệu gồm tập thuộc tính và nhãn.

Label: Nhãn của mẫu, đầu ra của bài toán phân lớp.

Training dataset: Tập dữ liệu huấn luyện, dùng để xây dựng mô hình

Testing dataset: Tập dữ liệu kiểm thử, dùng để kiểm tra độ chính xác của mô hình.

Feature extraction: ban đầu và xây dựng các giá trị bắt nguồn ( tính năng ) nhằm mục đích cung cấp thông tin và không dư thừa, tạo điều kiện cho các bước tổng quát hóa và học tập tiếp theo

Feature Selection: là quá trình lựa chọn một tập hợp con các đặc trưng có liên quan (các biến, các yếu tố dự báo) để sử dụng trong việc xây dựng mô hình. Các kỹ thuật lựa chọn tính năng được sử dụng vì một số lý do:

đơn giản hóa các mô hình để giúp các nhà nghiên cứu / người dùng giải thích chúng dễ dàng hơn

thời gian đào tạo ngắn hơn

để tránh lời nguyền về chiều không gian

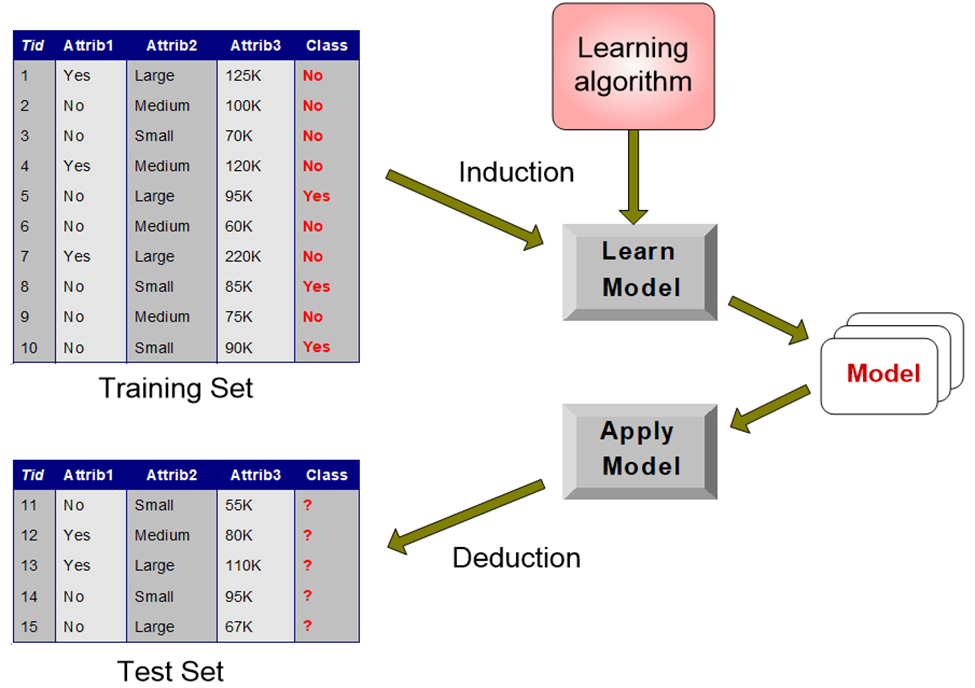
cải thiện khả năng tương thích của dữ liệu với một lớp mô hình học tập

mã hóa các đối xứng vốn có trong không gian đầu vào

Feature Reduction: là quá trình giảm số chiều dữ liệu trong một tính toán nặng về tài nguyên mà không làm mất thông tin quan trọng. Giảm số lượng chiều(thuộc tính) có nghĩa là số lượng công việc của máy tính dễ dàng hơn và nhanh hơn

Ground Truth : Nhãn thực tế của mẫu

### **Quy trình phân lớp**



Hình 1. 5 Quy trình phân lớp

Quy trình gồm 2 bước:

Xây dựng mô hình: mô tả một tập hợp các lớp định trước

- Mỗi mẫu được thuộc về một lớp được xác định trước, được xác định bởi thuộc tính nhãn lớp

- Tập hợp các mẫu được sử dụng để xây dựng mô hình là tập huấn luyện

- Mô hình được biểu diễ n dưới dạng các quy tắc phân loại, cây quyết định hoặc

công thức toán học

Sử dụng mô hình: để phân loại các mẫu chưa biết nhãn

- Ước tính độ chính xác của mô hình

+ Nhãn đã biết của mẫu thử nghiệm được so sánh với kết quả đã phân loại

từ mô hình

+ Tỷ lệ chính xác là tỷ lệ phần trăm các mẫu thử nghiệm đúng được phân loại bởi mô hình

+ Tập hợp kiểm tra độc lập với tập huấn luyện, nếu không sẽ xảy ra hiện tượng trùng quá mức

◦ Nếu độ chính xác được chấp nhận, mô hình được sử dụng để phân loại các mẫu không có nhãn.

### **Ví dụ**

* 1. Mô tả bài toán:

dự đoán xem bệnh nhân có bị bệnh tim hay không dự trên tập 2 thuộc tính cho sẵn

* 1. Mô tả dữ liệu:

Dữ liệu đầu vào gồm tập 2 thuộc tính và đầu ra (+) hoặc (-) tương đương với bệnh nhân có bệnh hoặc không bệnh.

* 1. Nhiệm vụ của bài toán phân lớp của ví dụ là làm gì?

Dự đoán bệnh nhân có bị bệnh tim hay không

## **Thuật toán K-NN**

### **Khái niệm**

KNN (K-Nearest Neighbors) là một trong những thuật toán học có giám sát đơn giản nhất được sử dụng nhiều trong khai phá dữ liệu và học máy. Ý tưởng của thuật toán này là nó không học một điều gì từ tập dữ liệu học (nên KNN được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán nhãn của dữ liệu mới.

Lớp (nhãn) của một đối tượng dữ liệu mới có thể dự đoán từ các lớp (nhãn) của k hàng xóm gần nó nhất.

### **Quy trình thực hiện Phân lớp dựa trên thuật toán KNN**

Bước 1:Ta có D là tập các điểm dữ liệu huấn luyện đã được gắn nhãn và A là dữ liệu chưa được phân loại.

Bước 2:Đo khoảng cách (Euclidian,..) từ A đến tất cả các mẫu dữ liệu trong D(Euclidian,..).

Bước 3: Chọn K (K là tham số mà bạn định nghĩa) khoảng cách nhỏ nhất.

Bước 4: Kiểm tra danh sách các lớp có khoảng cách ngắn nhất và đếm số lượng của mỗi lớp xuất hiện.

Lớp xuất hiện nhiều lần nhất là lớp của A

### **Viết mã giả**

Khởi Tạo tập dữ liệu huấn luyện T và dữ liệu Kiểm thử A

Tạo một mảng D để lưu trữ khoảng cách từ dữ liệu kiểm thử đến các phần tử của dữ liệu huấn luyện.

Duyệt Qua các phần tử T[i] tập dữ liệu huấn luyện

D[i]= Khoảng cách từ A đến T[i] (Euclidian,.).

Sắp xếp lại mảng D theo thứ tự từ bé đến lớn.

Khởi tạo tham số k là số hàng xóm gần nhất dùng để dự đoán nhãn.

Khởi tạo các biến đếm để đếm số lần xuất hiện của từng lớp

Duyệt mảng D i=0 đến k-1

Nếu D[i] thuộc lớp nào thì biến đếm lớp đó tăng lên 1

Lớp có số lần xuất hiện nhiều nhất là lớp được gán cho A.

### **Ví dụ minh họa:**

#### **Mô tả bài toán:**

Dự đoán xem bệnh nhân có bị bệnh tim hay không

#### **Mô tả dữ liệu:**

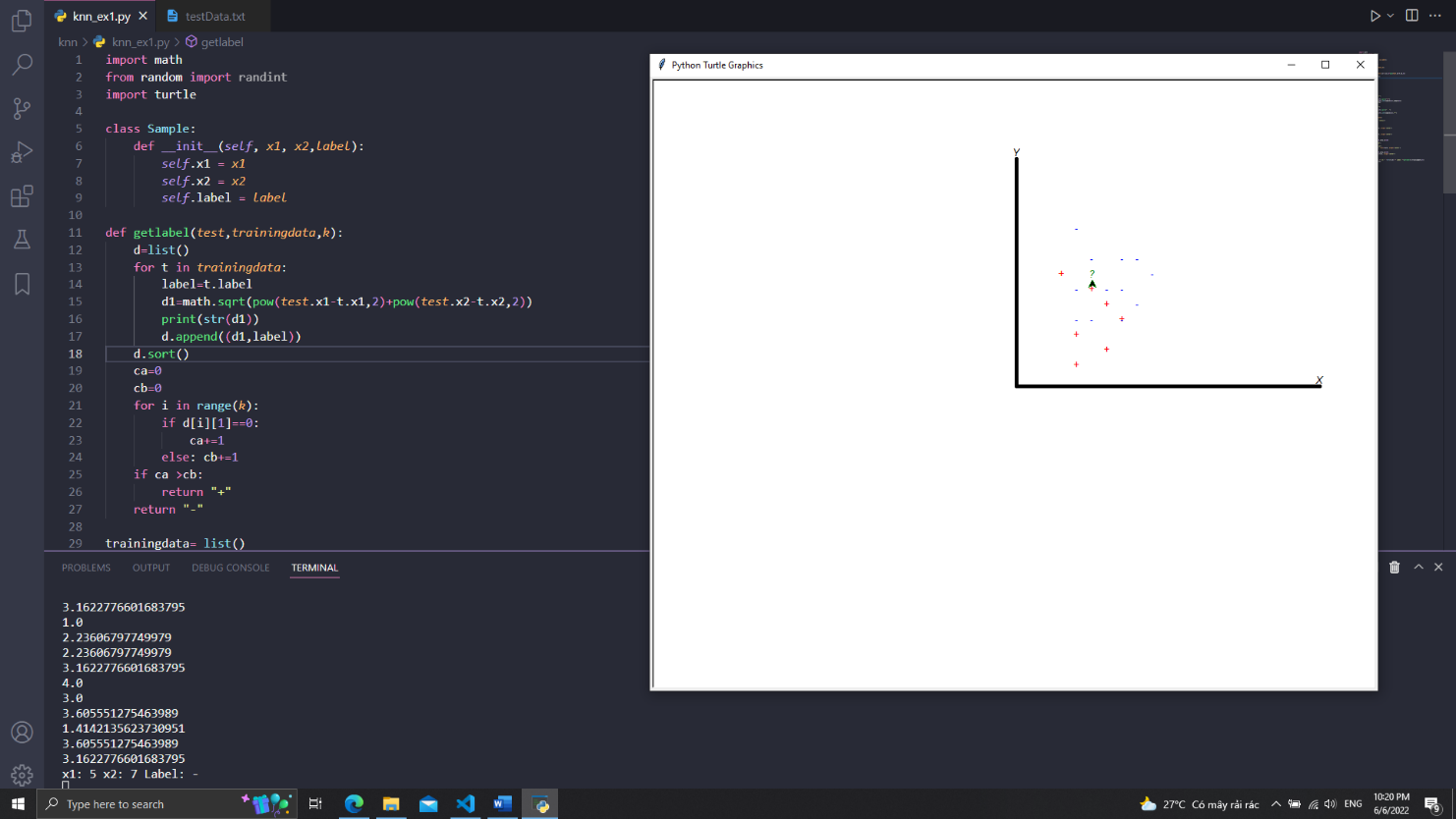
Dữ liệu đầu vào gồm tập 2 thuộc tính và đầu ra (+) hoặc (-) tương đương với bệnh nhân có bệnh hoặc không bệnh.

|  |
| --- |
| 4 3 +  3 7 +  7 4 +  4 1 +  6 5 +  5 6 +  3 7 +  6 2 +  4 6 -  4 4 -  5 8 -  7 8 -  7 6 -  4 10 -  9 7 -  5 4 -  8 5 -  6 6 -  7 4 -  8 8 - |
| **5 7 ?** |

#### **Thực hiện tính tay**

|  |  |
| --- | --- |
|  | Khoảng cách |
| 4 3 +  3 7 +  7 4 +  4 1 +  6 5 +  5 6 +  3 7 +  6 2 +  4 6 -  4 4 -  5 8 -  7 8 -  7 6 -  4 10 -  9 7 -  5 4 -  8 5 -  6 6 -  7 4 -  8 8 - | 4.123105625617661  2.0  3.605551275463989  6.082762530298219  2.23606797749979  1.0  2.0  5.0990195135927845  1.4142135623730951  3.1622776601683795  1.0  2.23606797749979  2.23606797749979  3.1622776601683795  4.0  3.0  3.605551275463989  1.4142135623730951  3.605551275463989  3.1622776601683795 |
| **5 7 ?** | **=>Label: -** |

### **Viết chương trình (source code) cho bài toán được lấy minh họa ở phần 4.**



Hình 1. 6 chương trình (source code) cho bài toán được lấy minh họa ở phần 4.

### **Trình bày Ưu/Nhược điểm của thuật toán KNN**

* Ưu điểm:

Thuật toán đơn giản, dễ dàng triển khai.

Độ phức tạp tính toán nhỏ.

Xử lý tốt với tập dữ liệu nhiễu

* Nhược điểm:

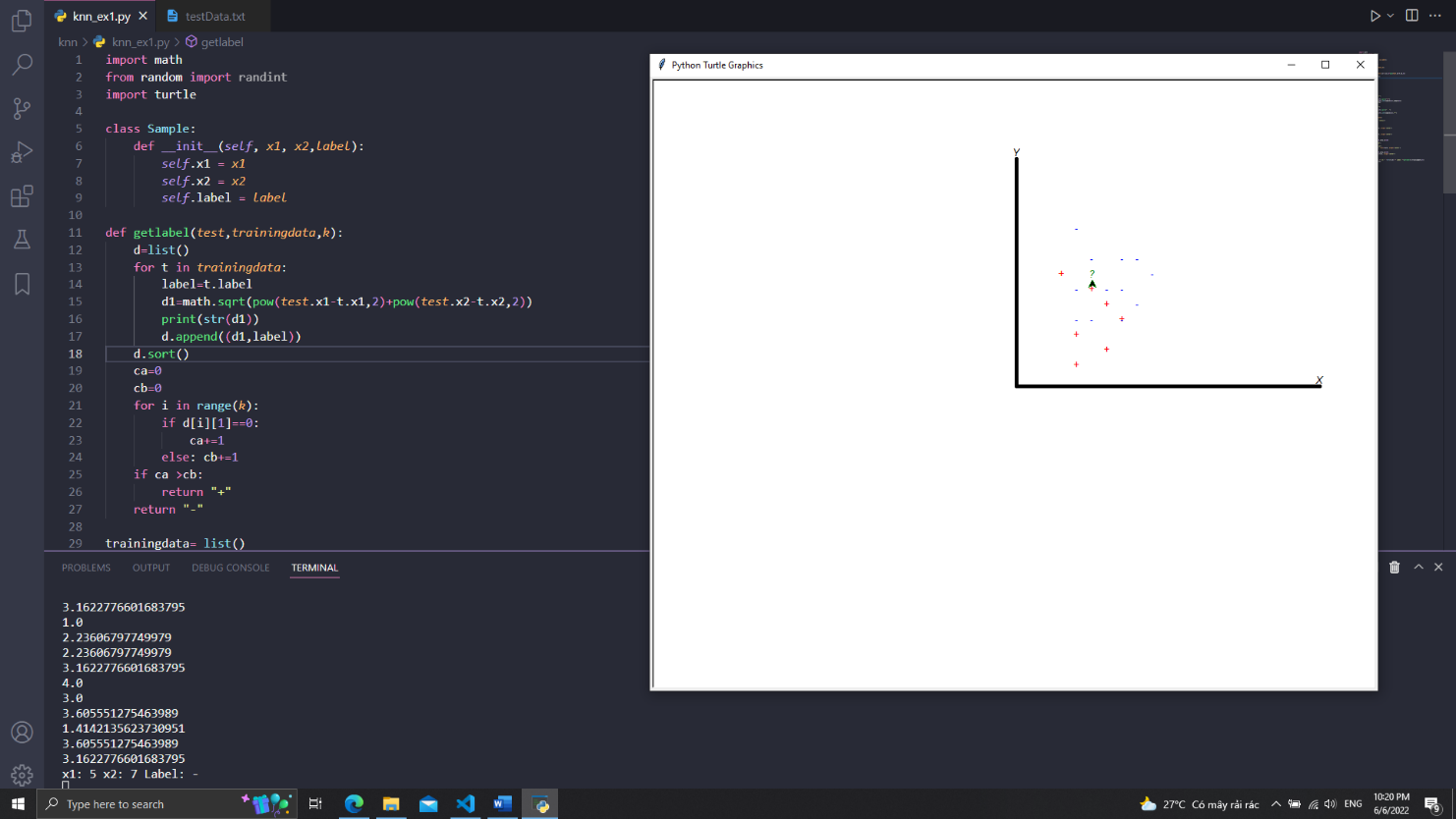
Với K nhỏ dễ gặp nhiễu dẫn tới kết quả đưa ra không chính xác

Cần nhiều thời gian để thực hiện do phải tính toán khoảng cách với tất cả các đối tượng trong tập dữ liệu.

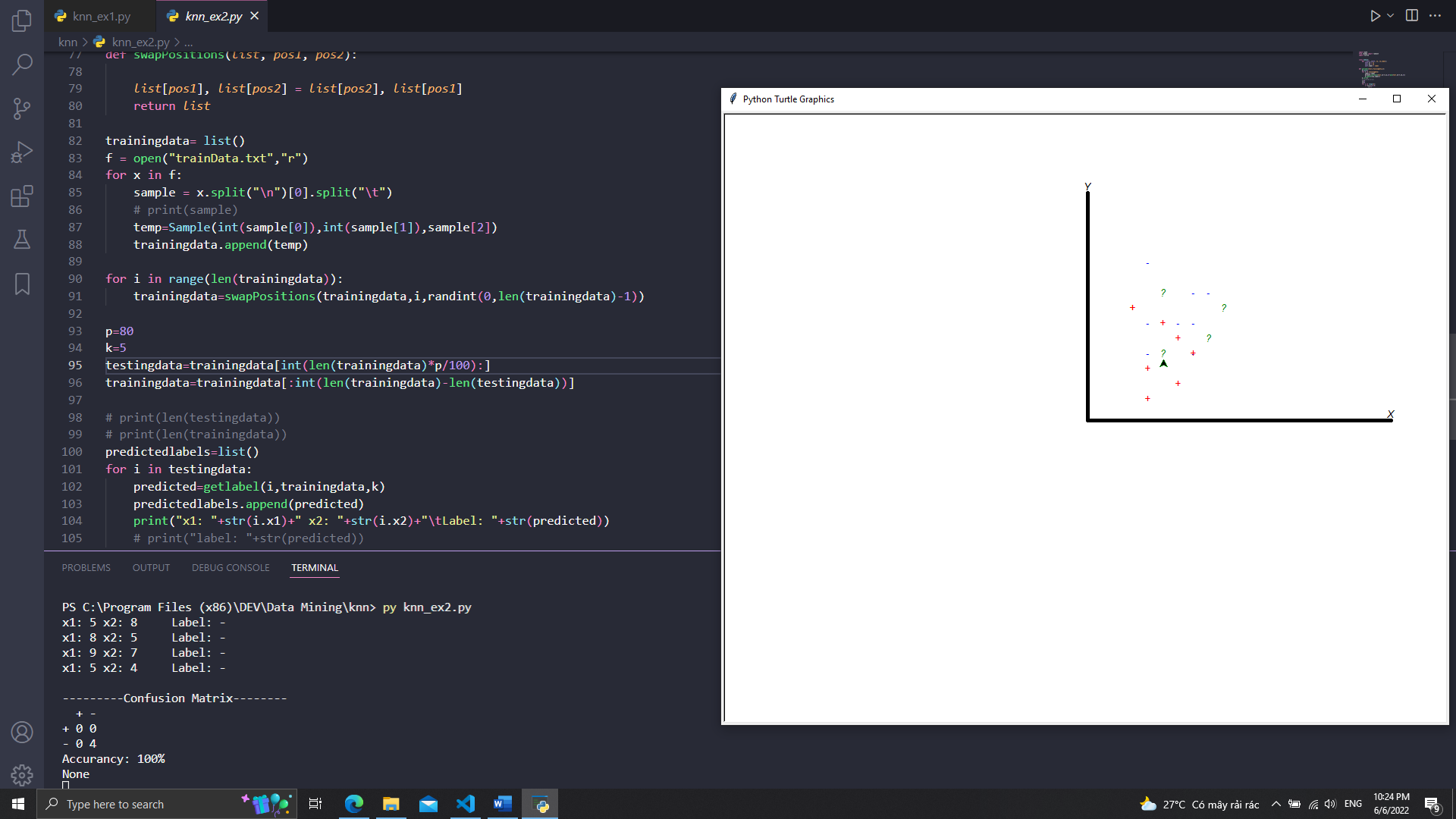
Cần chuyển đổi kiểu dữ liệu thành các yếu tố định tính.

### **Ứng dụng thực tế: Nghiên cứu ứng dụng thuật toán KNN trong việc chẩn đoán bệnh tim của bệnh nhân. Thực hiện theo yêu cầu như sau:**

* 1. **Tự viết code:** 
     1. Xác định tình trạng bệnh cho một bệnh nhân chưa biết dựa trên bộ dữ liệu huấn luyện cho sẵn.



Hình 1. 7 Source Code Knn

* + 1. Chia dữ liệu đã được cho sẵn: trong đó 80% dữ liệu là dùng cho để huấn luyện và 20% còn lại để kiểm thử

Hình 1. 8 Source Code Knn

## **Thuật toán Naïve Bayes:**

1. **Khái niệm**

Thuật toán Bayes là thuật toán phân loại theo xác suất dựa trên việc áp dụng định lý Bayes với các giả định độc lập giữa các đối tượng. Chúng là một trong những mô hình mạng Bayes đơn giản nhất, nhưng cùng với ước tính mật độ hạt nhân, chúng có thể đạt được mức độ chính xác cao.Quy trình thực hiện Phân lớp dựa trên thuật toán KNN

### **Viết mã giả**

### **Ví dụ minh họa:**

#### **Mô tả bài toán:**

Dự đoán thời tiết

#### **Mô tả dữ liệu:**

Dữ liệu gồm 4 chiều(thuộc tính) Outlook, Temp,Humidity ,Windy

#### **3.3. Thực hiện tính tay**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Outlook** | **Temp** | **Humidity** | **Windy** | **Play** |
| 1 | Sunny | Hot | High | False | No |
| 2 | Sunny | Hot | High | True | No |
| 3 | Overcast | Hot | High | False | Yes |
| 4 | Rainy | Mild | High | False | Yes |
| 5 | Rainy | Cool | Normal | False | Yes |
| 6 | Rainy | Cool | Normal | True | No |
| 7 | Overcast | Cool | Normal | True | Yes |
| 8 | Sunny | Mild | High | False | No |
| 9 | Sunny | Cool | Normal | False | Yes |
| 10 | Rainy | Mild | Normal | False | Yes |
| 11 | Sunny | Mild | Normal | True | Yes |
| 12 | Overcast | Mild | High | True | Yes |
| **ID** | **Outlook** | **Temp** | **Humidity** | **Windy** | **Play** |
| 13 | Overcast | Hot | Normal | False | Yes |
| 14 | Rainy | Mild | High | True | No |
| **15** | **Sunny** | **Cool** | **High** | **True** | **???** |

Kết quả tính tay:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Outlook** | | | **Temp** | | | **Humidity** | | | **Windy** | | | **Play** | |
|  | Yes | No |  | Yes | No |  | Yes | No |  | Yes | No | Yes | No |
| Sunny | 2 | 3 | Hot | 2 | 2 | High | 3 | 4 | False | 6 | 2 | 9 | 5 |
| Overcast | 4 | 0 | Mild | 4 | 2 | Normal | 6 | 1 | True | 3 | 3 |  |  |
| Rainy | 3 | 2 | Cool | 3 | 1 |  |  |  |  |  |  |  |  |
| **Sunny** | **2/9** | **3/5** | Hot | **2/9** | **2/5** | **High** | **3/9** | **4/5** | **False** | **6/9** | **2/5** | **9/14** | **5/14** |
| **Overcast** | **4/9** | **0/5** | Mild | **4/9** | **2/5** | **Normal** | **6/9** | **1/5** | **True** | **3/9** | **3/5** |  |  |
| **Rainy** | **3/9** | **2/5** | Cool | **3/9** | **1/5** |  |  |  |  |  |  |  |  |

P(X|Play=Yes) = P(Outlook = Sunny | Play = yes)

× P(Temp = Cool | Play = yes)

× P(Humidity = High | Play = yes)

× P(Windy = High | Play = yes)

= 2/9 × 3/9 × 3/9 × 3/9= 0.008230452675

Tương tự:

P(X| Play = No) = 3/5 × 1/5 × 4/5 × 3/5= 0.0576

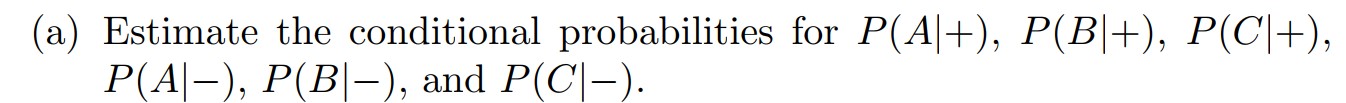
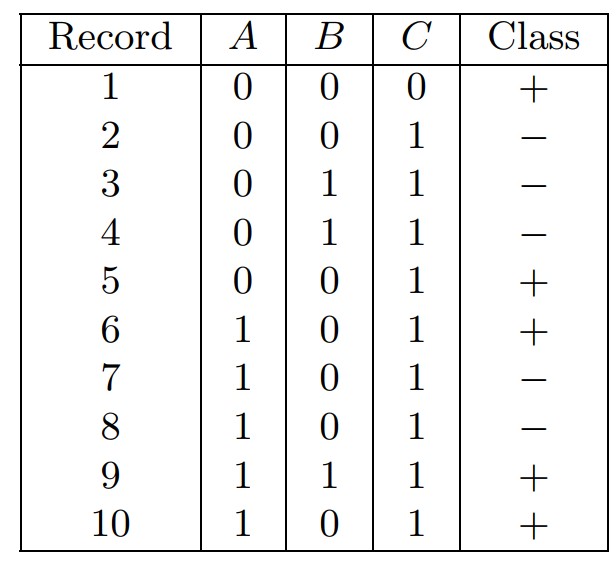
Để tìm ra nhãn(class) ta tính:

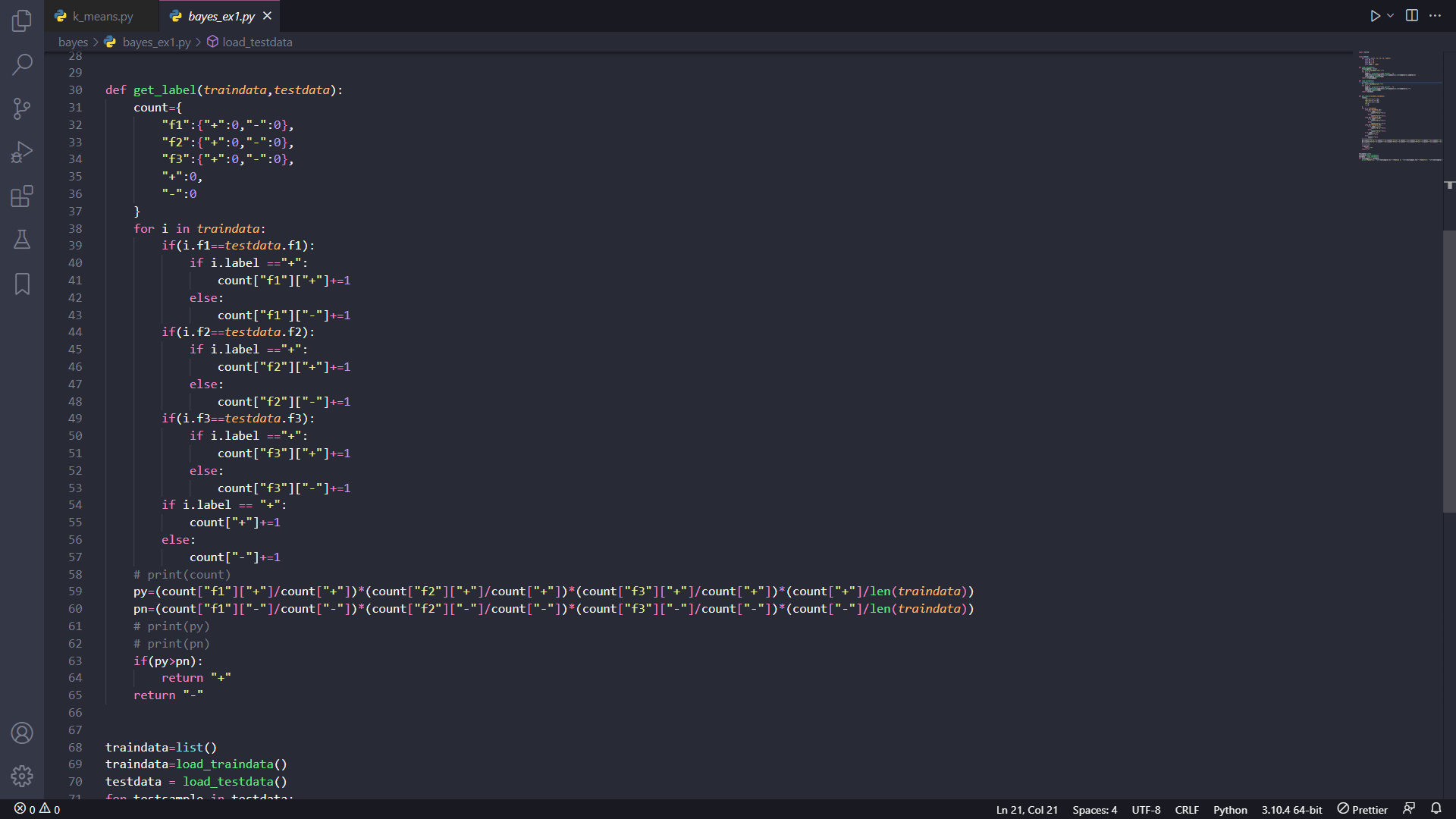
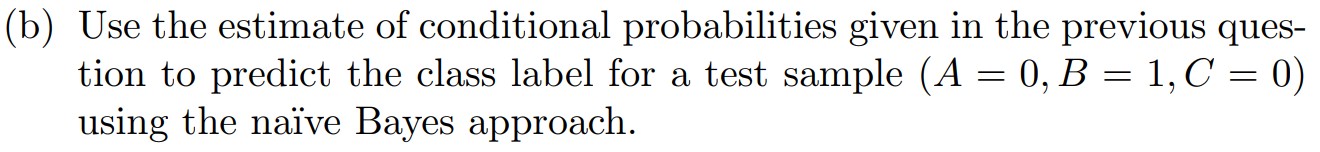
P(X| Play=Yes)P(Play = yes) = 0.008230452675× 9/14 = 0.005291005291

P(X| Play = No)P(Play = no) = 0.0576 × 5/14 = 0.2057142857

Vì Vậy, thuật toán Bayes sẽ dự đoán **Play=No** đối với mẫu X.

1. **Chương trình (source code)**





Hình 1. 9 Source Code Bayes

### **Ưu,Nhược điểm của thuật toán Bayes**

* **Ưu điểm**

Giả định độc lập: hoạt động tốt cho nhiều bài toán/miền sữ liệu và ứng dụng.

Đơn giản nhưng đủ tốt để giải quyết nhiều bài toán như phân lớp văn bản, lọc spam,..

Cho phép kết hợp tri thức tiền nghiệm (prior knowledge) và dữ liệu quan sát được (obserwed data).

Tốt khi có sự chệnh lệch số lượng giữa các lớp phân loại.

Huấn luyện mô hình (ước lượng tham số) dễ và nhanh.

* **Nhược điểm**

Giả định độc lập (ưu điểm cũng chính là nhược điểm)

Hầu hết các trường hợp thực tế trong đó có các thuộc tính trong các đối tượng thường phụ thuộc lẫn nhau.

Mô hình không được huẩn luyện bằng phượng pháp tối ưu mạnh và chặt chẽ.

Tham số mủa mô hình là các ước lượng xác suất điều kiện đơn lẻ.

Không tính đến sự tương tác giữa các ước lượng này.

## **Thuật toán Decision tree:**

1. **Khái niệm/ Ý tưởng**

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.

1. **Quy trình thực hiện**

Bước 1: Bắt đầu cây với nút gốc, S nói rằng, nút này chứa tập dữ liệu hoàn chỉnh.

Bước 2: Tìm thuộc tính tốt nhất trong tập dữ liệu bằng cách sử dụng Phép đo lựa chọn thuộc tính (ASM).

Bước 3: Chia S thành các tập con chứa các giá trị có thể có cho các thuộc tính tốt nhất.

Bước 4: Tạo nút cây quyết định chứa thuộc tính tốt nhất.

Bước 5: Tạo một cách đệ quy cây quyết định mới bằng cách sử dụng các tập con của tập dữ liệu đã tạo ở bước -3. Tiếp tục quá trình này cho đến khi đạt đến một giai đoạn mà bạn không thể phân loại thêm các nút và được gọi là nút cuối cùng là nút lá.

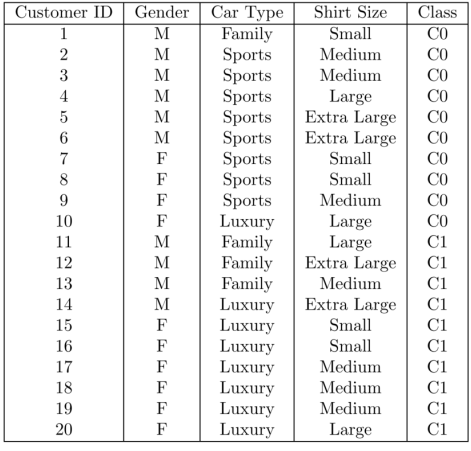
### **Viết mã giả**

### **Ví dụ minh họa**

4.1. Mô tả bài toán; Dự đoán lớp của mẫu

4.2. Mô tả dữ liệu: Dữ liệu 3 chiều gồm 3 thuộc tính(feature) Gender,Car Type,Shirt Size

4.3. Thực hiện tính tay



Tính chỉ số Gini

Gini(S)= 1 -((∑(p\_i )^2)┬(i=1))┴n

vì bộ traning này có 20 bản ghi và hai lớp có thể có tần số tương đối và (1-p), tương ứng. Trong trường hợp này, C0 và C1 có cùng tần số tương đối (p = 1 – p = 1/2):

Gini = 1 – (p)¬2 - ( 1 – p)2 = 2p (1 - p)=2p2

= 2 × 1/4= 1/2 = 0.5

Tính Gin index cho thuộc tính Customer ID

Gini(S)= -∑\_(i=1)^20▒0 = 0

Vì mỗi bản ghi của thuộc tính này không có tính lặp dữ liệu nên mục gin sẽ là 0

Tính Gin index cho thuộc tính Gender:

P(M) = 10/20 = 0.5 P(F) = 10/20 = 0.5

P(M|C0) = 6/10 = 0.6

P(M|C1) = 4/10 = 0.4

Gini(M) = 1 – ((0.6)2 + (0.4)2) = 0.48

P(F|C0) = 4/10 = 0.4

P(F|C1) = 6/10 = 0.6

Gini(F) = 1 – ((0.4)2 + (0.6)2) = 0.48

Gini(Gender) = 0.5 × 0.48 + 0.5 × 0.48 = 0.48

Tính Gin index cho thuộc tính Car type

P(family) = 4/20 = 0.2; P(luxury) = 8/20 = 0.4; P(sports) = 8/20 = 0.4

P(family|C0) = 1/4 = 0.25

P(family|C1) = 3/4 = 0.75

+ Gini(family) = 1 – ((0.25)2 + (0.75)2) = 0.375

P(luxury|C0) = 1/8 = 0.125

P(luxury|C1) = 7/8 = 0.875

+ Gini(luxury) = 1 – ((0.125)2 + (0.875)2) = 0.219

P(sports|C0) = 8/8 = 1

P(sports|C1) = 0/8 = 0

+ Gini(sports) = 1 – ((1)2 + (0)2) = 0

Gini(Car type) = 0.2 × 0.375 + 0.8 × 0.218 + 0.8 0 = 0.163

Tính Gin index cho thuộc tính Shirt size

P(small) = 5/20 = 0.25 ; P(medium) = 7/20 = 0.35;

P(large) = 4/20 = 0.2 ; P(extra large) = 4/20 = 0.2

P(small|C0) = 3/5 = 0.6

P(small|C1) = 2/5 = 0.4

+ Gini(small) = 1 – ((0.6)2 + (0.4)2) = 0.48

P(medium|C0) = 3/7 = 0.429

P(medium|C1) = 4/7 = 0.571

+ Gini(medium) = 1 – ((0.429)2 + (0.571)2) = 0.49

P(large|C0) = 2/4 = 0.5

P(large|C1) = 2/4 = 0.5

+ Gini(large) = 1 – ((0.5)2 + (0.5)2) = 0.5

P(extra large|C0) = 2/4 = 0.5

P(extra large|C1) = 2/4 = 0.5

+ Gini(extra large) = 1 – ((0.5)2 + (0.5)2) = 0.5

 Gini(Shirt size)= 0.25×0.48 + 0.35×0.49 + 0.2×0.5 + 0.2 ×0.5

= 0.4915

### **Ưu/Nhược điểm của thuật toán Decision tree:**

* **Ưu điểm của cây quyết định**

Có thể hiểu đơn giản là nó tuân theo cùng một quy trình mà con người tuân theo khi đưa ra bất kỳ quyết định nào trong cuộc sống thực.

Nó có thể rất hữu ích để giải quyết các vấn đề liên quan đến quyết định.

Nó giúp bạn suy nghĩ về tất cả các kết quả có thể xảy ra cho một vấn đề.

Yêu cầu làm sạch dữ liệu ít hơn so với các thuật toán khác.

* **Nhược điểm của cây quyết định**

Cây quyết định chứa rất nhiều lớp, điều này làm cho nó trở nên phức tạp.

Nó có thể có vấn đề trang bị quá mức, có thể được giải quyết bằng cách sử dụng thuật toán Rừng ngẫu nhiên.

Đối với nhiều nhãn lớp hơn, độ phức tạp tính toán của cây quyết định có thể tăng lên.

## **Phân cụm**

### **Khái niệm**

Phân cụm là quá trình phân vùng một tập hợp các đối tượng dữ liệu (hoặc các quan sát) thành các tập con. Mỗi tập hợp con là một cụm, sao cho các đối tượng trong một cụm tương tự với nhau, nhưng không giống với các đối tượng trong các cụm khác. Tập hợp các cụm kết quả từ phân tích cụm có thể được gọi là phân nhóm. Trong bối cảnh này, các phương pháp phân cụm khác nhau có thể tạo ra các chuỗi khác nhau trên cùng một tập dữ liệu. Các phân vùng không được thực hiện bởi con người, nhưng bởi thuật toán phân cụm. Do đó, clustering hữu ích ở chỗ nó có thể dẫn đến việc phát hiện ra các nhóm chưa biết trước đây trong dữ liệu.

### **Mục đích của Phân cụm**

Trong nhiều lĩnh vực, phân tích cụm được sử dụng, bao gồm nhận dạng mẫu, phân tích hình ảnh, truy xuất thông tin, tin sinh học, nén dữ liệu, đồ họa máy tính và học máy.

### **Phân cụm và Phân lớp**

|  |  |  |
| --- | --- | --- |
|  | Phân Lớp | Phân Cụm |
| Khái niệm | Phân loại các lớp hoặc khái niệm dữ liệu | Phân cụm là quá trình phân vùng một tập hợp các đối tượng dữ liệu |
| Cơ chế học | Học có giám sát | Học tập không có giám sát |
| Mục đích | Tổ chức được các sự vật, vật chất, hiện tượng, con người, động vật, thực vật,… thành các lớp | Nhận dạng mẫu, phân tích hình ảnh, truy xuất thông tin, tin sinh học, nén dữ liệu, đồ họa máy tính và học máy. |

## **Thuật toán K-means:**

1. **Khái niệm/ Ý tưởng**

Thuật toán phân cụm k-means là một phương pháp được sử dụng trong phân tích tính chất cụm của dữ liệu. Nó đặc biệt được sử dụng nhiều trong khai phá dữ liệu và thống kê. Nó phân vùng dữ liệu thành k cụm khác nhau.

1. **Quy trình thực hiện Phân lớp dựa trên thuật toán K-means**

Bước 1:Khởi tạo K điểm dữ liệu trong bộ dữ liệu và tạm thời coi nó là tâm của các cụm dữ liệu của chúng ta.

Bước 2: Với mỗi điểm dữ liệu trong bộ dữ liệu, tâm cụm của nó sẽ được xác định là 1 trong K tâm cụm gần nó nhất.

Bước 3: Sau khi tất cả các điểm dữ liệu đã có tâm, tính toán lại vị trí của tâm cụm để đảm bảo tâm của cụm nằm ở chính giữa cụm.

Bước 2 và bước 3 sẽ được lặp đi lặp lại cho tới khi vị trí của tâm cụm không thay đổi hoặc tâm của tất cả các điểm dữ liệu không thay đổi.

### **Viết mã giả**

* Đầu vào:

k: số lượng cụm,

D: tập dữ liệu chứa n đối tượng.

Đầu ra: Một tập hợp k cụm.

* Phương pháp:

(1) Tự ý chọn k đối tượng ở D làm đối tượng đại diện ban đầu hoặc hạt giống;

(2) lặp lại

(3) gán từng đối tượng còn lại vào cụm với đối tượng đại diện gần nhất;

(4) chọn ngẫu nhiên một đối tượng không đại diện, hoặc địa chỉ;

(5) tính tổng chi phí, S, của việc hoán đổi đối tượng đại diện, oj

, với orandom;

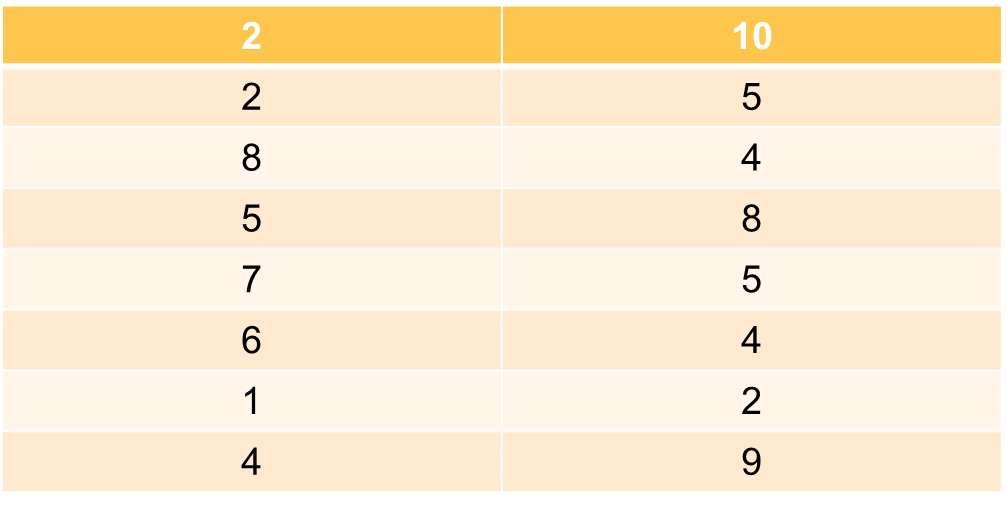
(6) nếu S <0 thì hoán đổi oj với orandom để tạo thành tập k đối tượng đại diện mới;

(7) cho đến khi không thay đổi;

### **Ví dụ minh họa**

4.1. Mô tả bài toán: Phân 8 mẫu thành 3 cụm

4.2. Mô tả dữ liệu: Dữ liệu 2 chiều(Feature)



4.3. Thực hiện tính tay

Tâm ban đầu: (2,10),(2,5),(8,4)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **x** | **y** | **(2,10)** | **(2,5)** | **(8,4)** | **Cụm** |
| 2 | 10 | 0.0 | 5.0 | 8.48 | 1 |
| 2 | 5 | 5.0 | 0.0 | 6.08 | 2 |
| 8 | 4 | 8.48 | 6.08 | 0.0 | 3 |
| 5 | 8 | 3.6 | 4.24 | 5.0 | 1 |
| 7 | 5 | 7.07 | 5.0 | 1.41 | 3 |
| 6 | 4 | 7.21 | 4.12 | 2.0 | 3 |
| 1 | 2 | 8.06 | 3.16 | 7.28 | 2 |
| 4 | 9 | 2.23 | 4.47 | 6.40 | 1 |

Cụm 1: [(2, 10), (5, 8), (4, 9)]

Cụm 2: [(2, 5), (1, 2)]

Cụm 3: [(8, 4), (7, 5), (6, 4)]

Tâm sau khi cập nhật: (3.66, 9.0) ,(1.5, 3.5), (7.0, 4.33)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **x** | **y** | **(3.66, 9.0)** | **(1.5, 3.5)** | **(7.0, 4.33)** | **Cụm** |
| 2 | 10 | 1.94 | 6.51 | 7.55 | 1 |
| 2 | 5 | 4.33 | 1.58 | 5.04 | 2 |
| 8 | 4 | 6.61 | 6.51 | 1.05 | 3 |
| 5 | 8 | 1.66 | 5.7 | 4.17 | 1 |
| 7 | 5 | 5.20 | 5.7 | 0.66 | 3 |
| 6 | 4 | 5.51 | 4.52 | 1.05 | 3 |
| 1 | 2 | 7.49 | 1.58 | 6.43 | 2 |
| 4 | 9 | 0.33 | 6.04 | 5.54 | 1 |

Cụm 1: [(2, 10), (5, 8), (4, 9)]

Cum 2: [(2, 5), (1, 2)]

Cum 3: [(8, 4), (7, 5), (6, 4)]

Tâm sau khi cập nhật: (3.66, 9.0) ,(1.5, 3.5), (7.0, 4.33)

=> Dừng, đây là kết quả cho bài toán

1. **Ưu/Nhược điểm của thuật toán K-means**

Nhược điểm

Chúng ta cần biết số lượng cluster cần clustering

Nghiệm cuối cùng phụ thuộc vào các centers được khởi tạo ban đầu

Các cluster cần có só lượng điểm gần bằng nhau

Các cluster cần có dạng hình tròn

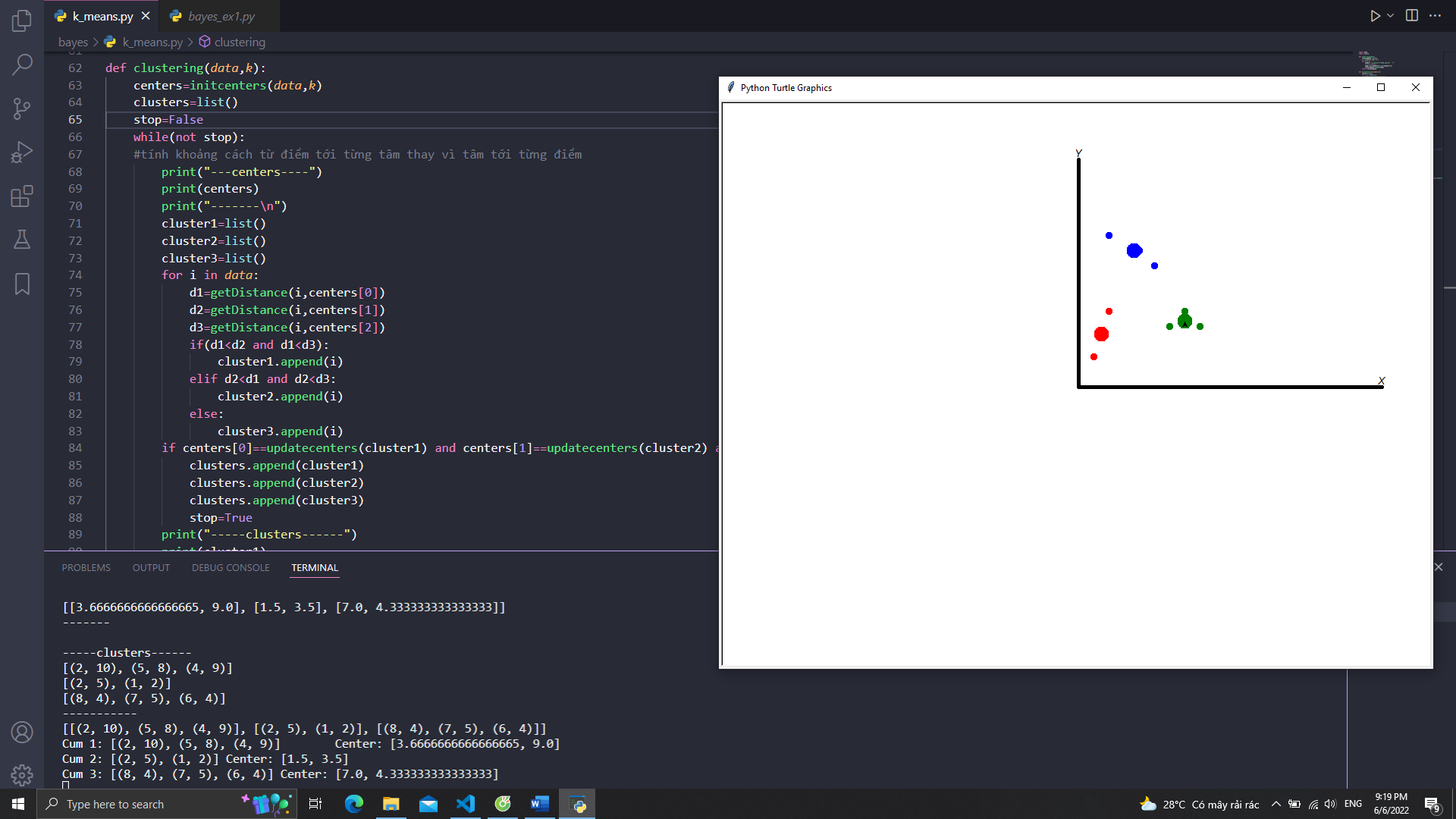
Khi một cluster nằm phía trong 1 cluster khác

## **Luật kết hợp**

### **Khái niệm**

Học quy tắc kết hợp là một phương pháp học máy dựa trên quy tắc để khám phá các mối quan hệ thú vị giữa các biến trong cơ sở dữ liệu lớn. Nó nhằm xác định các quy tắc mạnh được phát hiện trong cơ sở dữ liệu bằng cách sử dụng một số thước đo mức độ thú vị. Trong bất kỳ giao dịch nhất định nào với nhiều mặt hàng, quy tắc kết hợp có nghĩa là khám phá các quy tắc xác định cách thức hoặc lý do tại sao một số mặt hàng được kết nối với nhau.

### **Viết chương trình (source code) cho bài toán được lấy minh họa ở phần 4.**



Hình 1. 10 Source Code K-means

### **Các miền ứng dụng**

### **Thuật toán APRIORI:**

* 1. Trình bày ý tưởng

Apriori sử dụng phương pháp tiếp cận "từ dưới lên", trong đó các tập hợp con thường xuyên được mở rộng từng mục một (bước được gọi là tạo ứng viên ) và các nhóm ứng viên được kiểm tra dựa trên dữ liệu. Thuật toán kết thúc khi không tìm thấy phần mở rộng thành công nào nữa. Apriori sử dụng tìm kiếm theo chiều rộng và cấu trúc cây Hash để đếm các tập hợp mục ứng viên một cách hiệu quả. Nó tạo ra các tập hợp chiều dài mục ứng viên từ các tập hợp chiều dài. Sau đó, nó sẽ loại bỏ những ứng cử viên có mẫu phụ không thường xuyên. Theo bổ đề đóng xuống, tập ứng viên chứa tất cả các tập mục có độ dài thường xuyên. Sau đó, nó sẽ quét cơ sở dữ liệu giao dịch để xác định các nhóm mặt hàng thường xuyên giữa các ứng viên.

* 1. Ví dụ minh họa

Ví dụ: Giả sử rằng mỗi hàng là một mẫu ung thư với sự kết hợp nhất định của các đột biến được đánh dấu bằng một ký tự trong bảng chữ cái. Ví dụ: một hàng có thể có {a, c} có nghĩa là hàng đó bị ảnh hưởng bởi đột biến "a" và đột biến "c".

**Bộ đầu vào**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **{a, b}** | **{đĩa CD}** | **{a, d}** | **{a, e}** | **{b, d}** | **{a, b, d}** | **{a, c, d}** | **{A B C D}** |

Bây giờ chúng ta sẽ tạo tập mục thường xuyên bằng cách đếm số lần xuất hiện của mỗi ký tự. Điều này còn được gọi là tìm các giá trị hỗ trợ. Sau đó, chúng tôi sẽ lược bỏ mục đã đặt bằng cách chọn một ngưỡng hỗ trợ tối thiểu. Đối với lần vượt qua thuật toán này, chúng tôi sẽ chọn 3.

**Giá trị hỗ trợ**

|  |  |  |  |
| --- | --- | --- | --- |
| **một** | **b** | **c** | **d** |
| 6 | 4 | 3 | 6 |

Vì tất cả các giá trị hỗ trợ đều từ ba trở lên nên không có sự cắt tỉa. Tập hợp mục thường xuyên là {a}, {b}, {c} và {d}. Sau đó, chúng tôi sẽ lặp lại quá trình bằng cách đếm các cặp đột biến trong tập hợp đầu vào.

**Giá trị hỗ trợ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **{a, b}** | **{AC}** | **{a, d}** | **{b, c}** | **{b, d}** | **{đĩa CD}** |
| 3 | 2 | 4 | 1 | 3 | 4 |

Bây giờ chúng ta sẽ đặt giá trị hỗ trợ tối thiểu của mình là 4 để chỉ còn lại {a, d} và {c, d} sau khi cắt bớt. Bây giờ chúng ta sẽ sử dụng tập hợp mục thường xuyên để tạo sự kết hợp của các bộ ba. Sau đó, chúng tôi sẽ lặp lại quy trình bằng cách đếm số lần xuất hiện của các bộ ba đột biến trong tập hợp đầu vào.

**Giá trị hỗ trợ**

|  |
| --- |
| **{a, c, d}** |
| 2 |

Vì chúng ta chỉ có một mục nên tập hợp các tổ hợp bộ bốn tiếp theo là trống nên thuật toán sẽ dừng lại.

* 1. Ưu/Nhược điểm của thuật toán

Apriori có một số hạn chế. Việc tạo ra ứng viên có thể dẫn đến các tập hợp ứng viên lớn. Ví dụ: tập hợp 1 mục thường xuyên 10 ^ 4 sẽ tạo ra tập hợp 2 mục ứng cử viên 10 ^ 7. Thuật toán cũng cần thường xuyên quét cơ sở dữ liệu, cụ thể là n + 1 lần quét trong đó n là độ dài của mẫu dài nhất. Apriori chậm hơn thuật toán Eclat. Tuy nhiên, Apriori hoạt động tốt so với Eclat khi tập dữ liệu lớn. Điều này là do trong thuật toán Eclat nếu tập dữ liệu quá lớn thì danh sách tid trở nên quá lớn so với bộ nhớ. FP-GROWTH vượt trội hơn Apriori và Eclat. Điều này là do thuật toán FP-GROWTH không có ứng viên tạo hoặc kiểm tra, sử dụng cấu trúc dữ liệu nhỏ gọn và chỉ có một lần quét cơ sở dữ liệu.

### **Thuật toán FP-GROWTH:**

* 1. Ý tưởng

Nén cơ sở dữ liệu vào cây FP-tree, chỉ giữ lại thông tin liên kết (kết hợp) của các hạng mục (tập phổ biến).

Chia cơ sở dữ liệu nén thành các cơ sở dữ liệu có điều kiện, mỗi cơ sở dư liệu được chia ra ứng với một hạng mục phổ biến và ta sẽ khai thác các cơ sở dữ liệu này một cách độc lập. Cho ví dụ minh họa: Phát biểu bài toán, mô tả dữ liệu và trình bày cách thực hiện.

* 1. Ưu/Nhược điểm của thuật toán

Số lượng khổng lồ các tập ứng viên được sinh ra

## **Đánh giá mô hình**

### **Các phương thức để xác định hiệu năng của mộ hình Phân lớp/Phân cụm**

* Accuracy
* Confusion matrix
* True/False Positive/Negative
* Precision và Recall

### **Phương pháp Confusion matrix**

* Định nghĩa

Ma trận lỗi là một bố cục bảng cụ thể cho phép hình dung hiệu suất của một thuật toán, thường là một thuật toán có giám sát. Mỗi hàng của ma trận đại diện cho các trường hợp trong một lớp thực tế trong khi mỗi cột đại diện cho các trường hợp trong một lớp được dự đoán hoặc ngược lại . Cái tên này xuất phát từ thực tế là nó giúp bạn dễ dàng nhận ra liệu hệ thống có đang nhầm lẫn giữa hai lớp hay không (tức là thường gắn nhãn sai cho một lớp khác).

* Các tham số:

Predicted Label: Nhãn được đưa ra bởi mô hình

Growth Truth: Nhãn thực tế của mẫu

### **Cách tính độ chính xác mô hình thông qua confusion matrix**

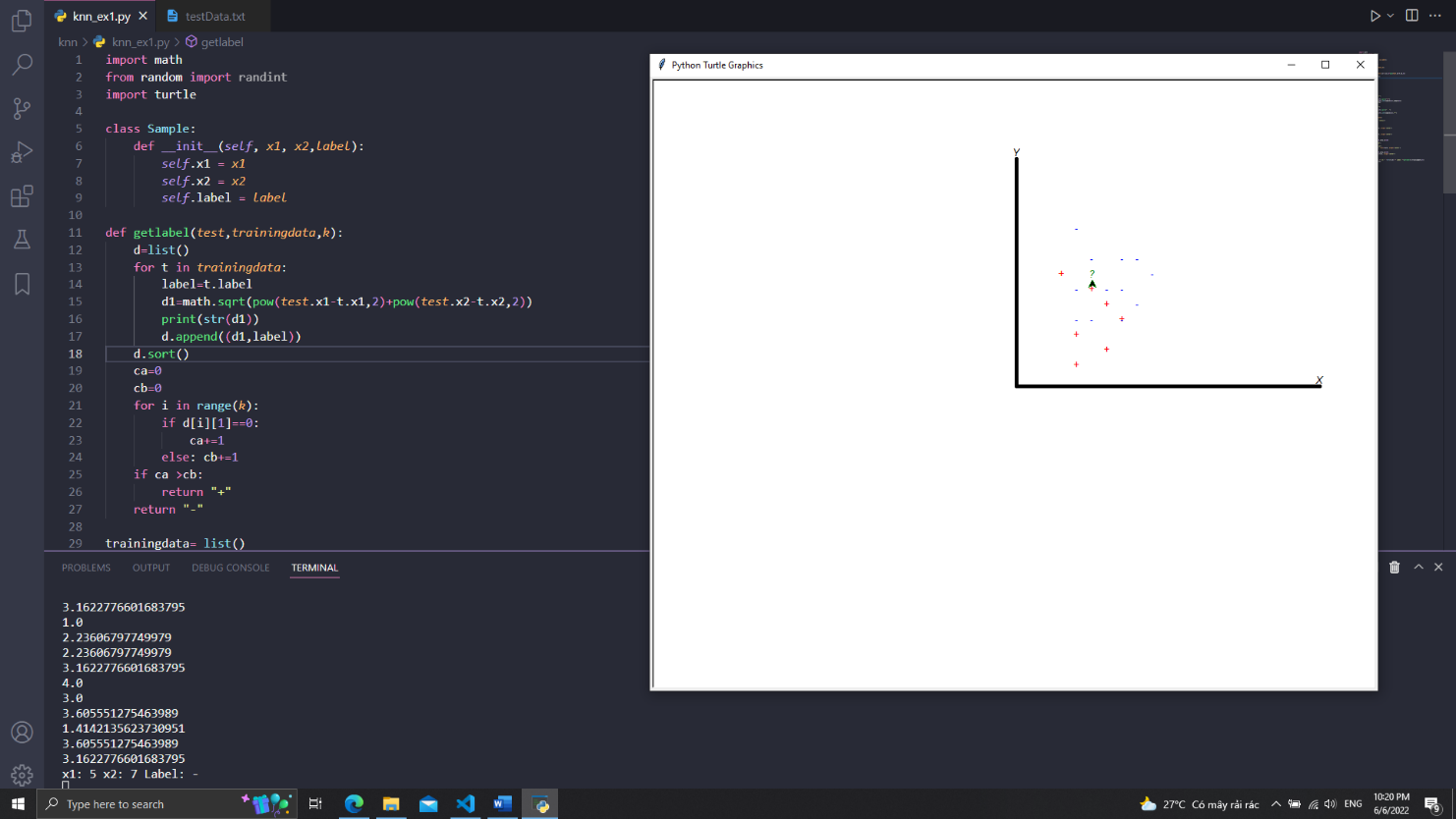
Accuracy

**PHẦN II: THỰC HÀNH**

# Diagram Description automatically generatedEx2: Given a collection dataset as following: Using KNN method

1. Classify the query point squery=(7,5) according to its 1-, 3-, 5-, and 9-nearest neighbors (using majority vote).
2. Using the training dataset in Ex2, dividing 80% for training and 20% for testing. Using kNN method for this dataset, how about classification accuracy with k = 1-9 (odd number) (using majority vote). Applying the confusion matrix to evaluate the model performance.

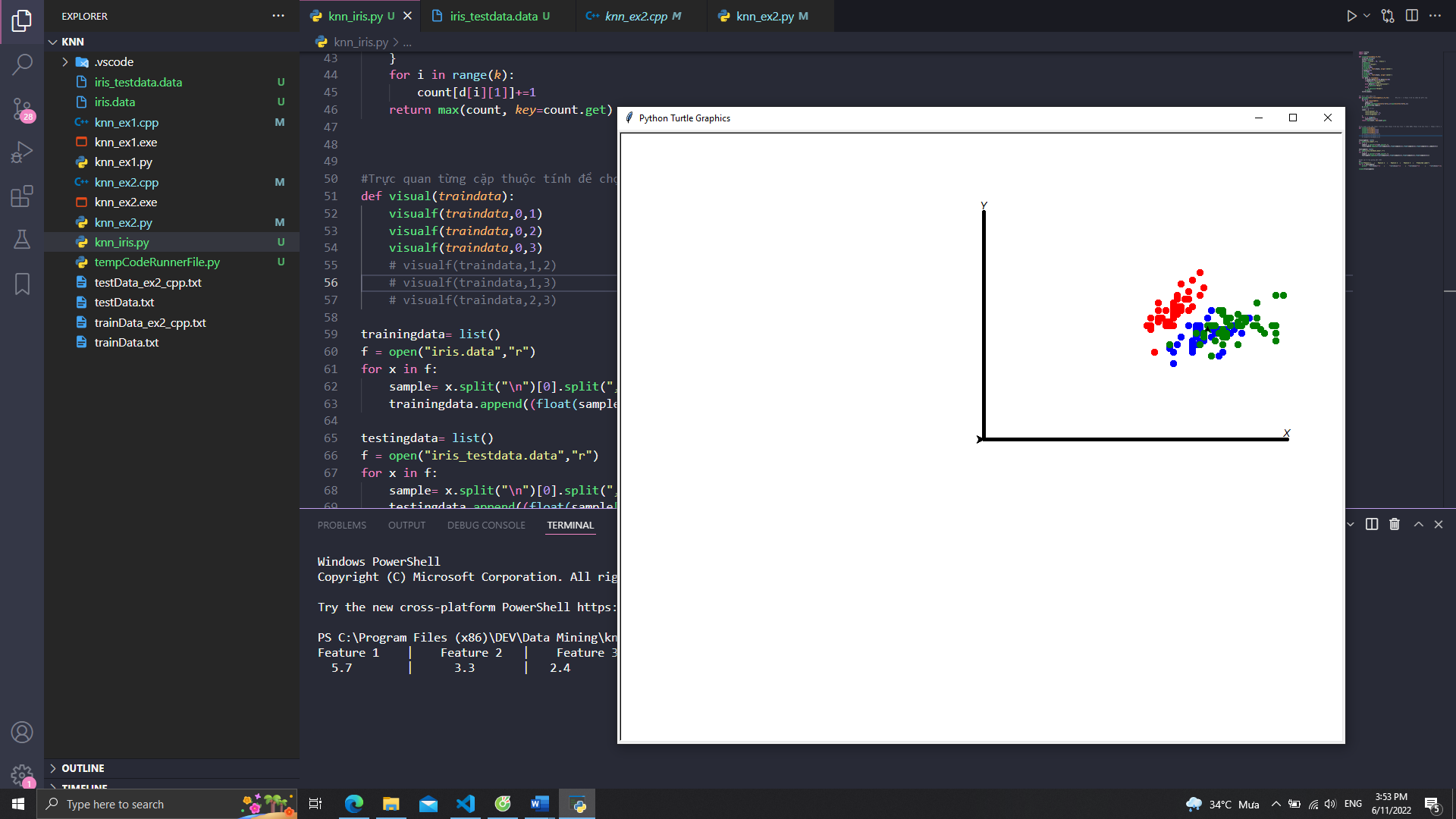
|  |  |
| --- | --- |
|  | Khoảng cách |
| 4 3 +  3 7 +  7 4 +  4 1 +  6 5 +  5 6 +  3 7 +  6 2 +  4 6 -  4 4 -  5 8 -  7 8 -  7 6 -  4 10 -  9 7 -  5 4 -  8 5 -  6 6 -  7 4 -  8 8 - | 4.123105625617661  2.0  3.605551275463989  6.082762530298219  2.23606797749979  1.0  2.0  5.0990195135927845  1.4142135623730951  3.1622776601683795  1.0  2.23606797749979  2.23606797749979  3.1622776601683795  4.0  3.0  3.605551275463989  1.4142135623730951  3.605551275463989  3.1622776601683795 |
| **5 7 ?** | **=>Label: -** |



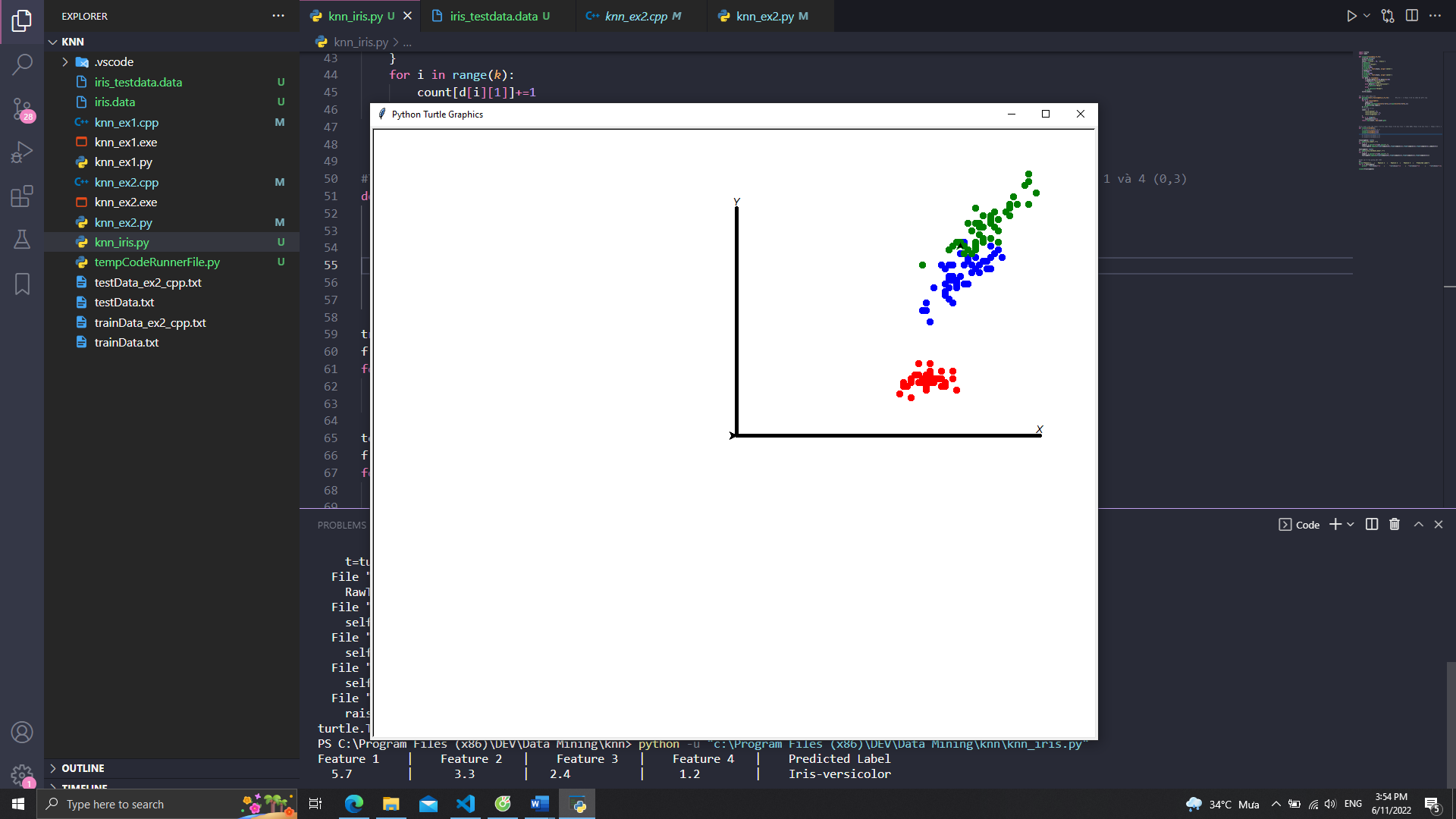
# Ex3: Applying KNN on real Iris dataset, show experimental results for classification problem on the two strongest features and full features. Discuss and conclude on those results. You can download dataset directly from

[**https://archive.ics.uci.edu/ml/datasets/iris**](https://archive.ics.uci.edu/ml/datasets/iris)

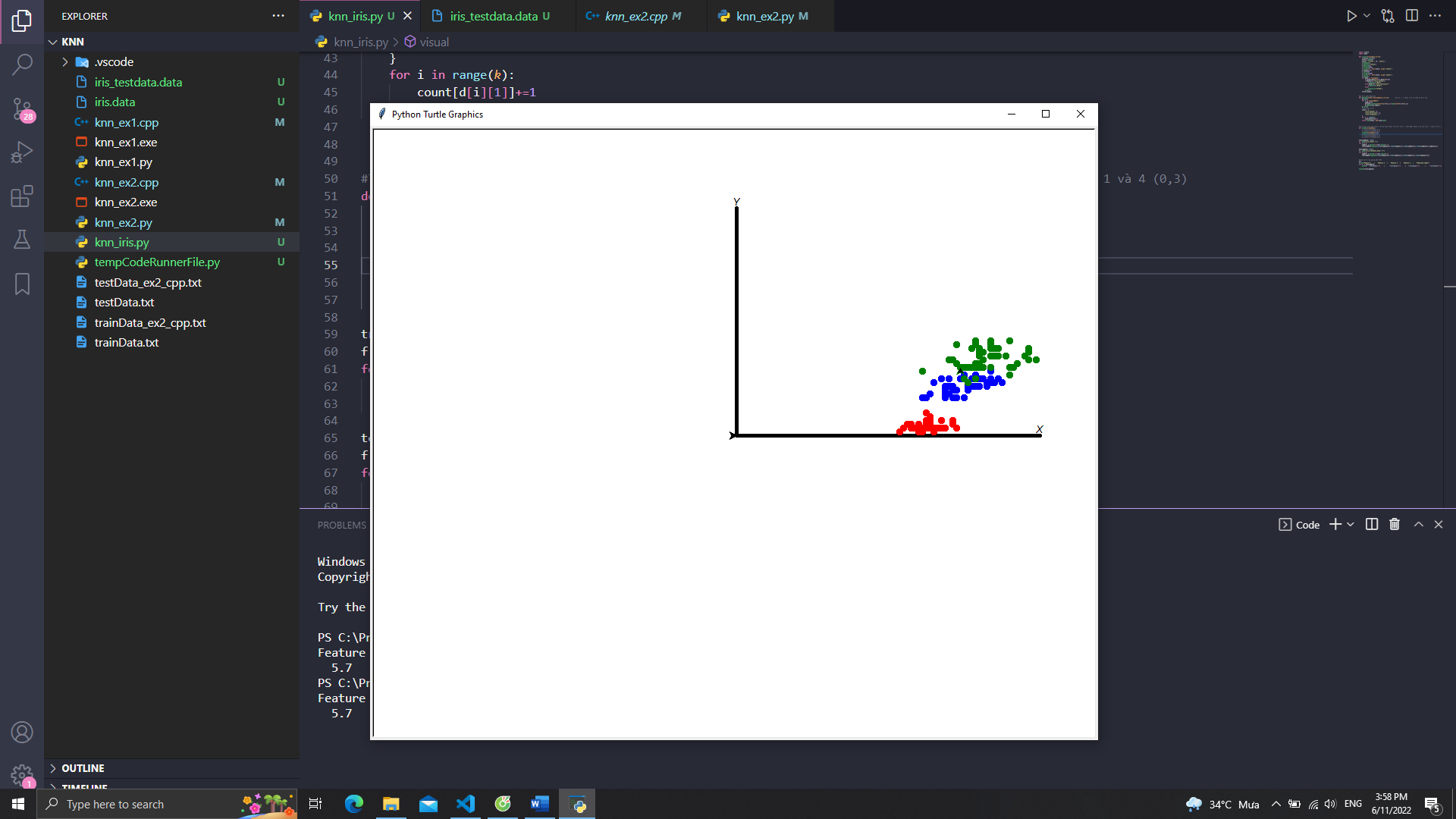
*Note: To choose the two strongest features, you can visualize a couple of features in 2D via using scatter plot and get the best one.*



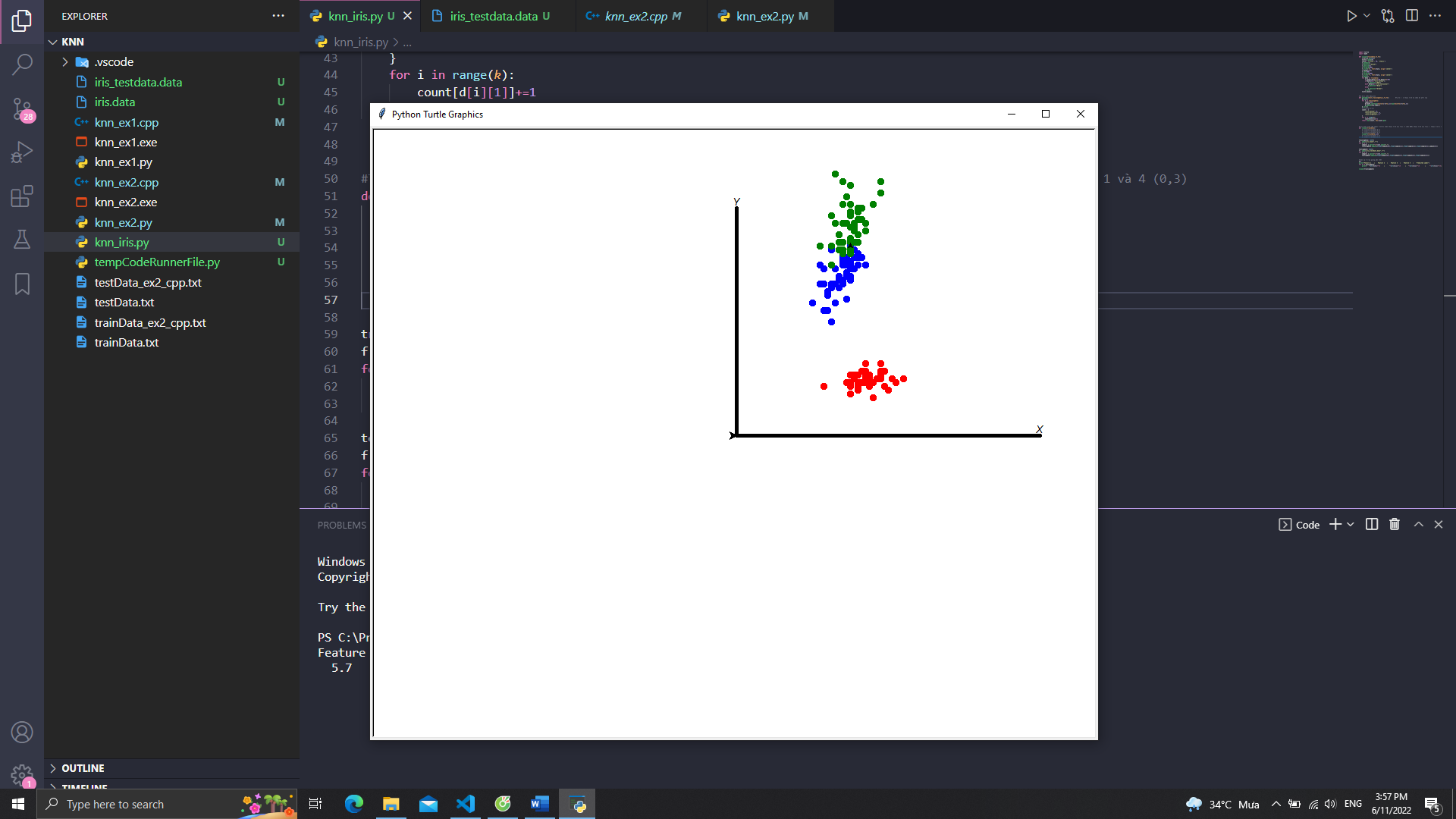
Hình 2. 1Trực quan 2 thuộc tính 1 và 2



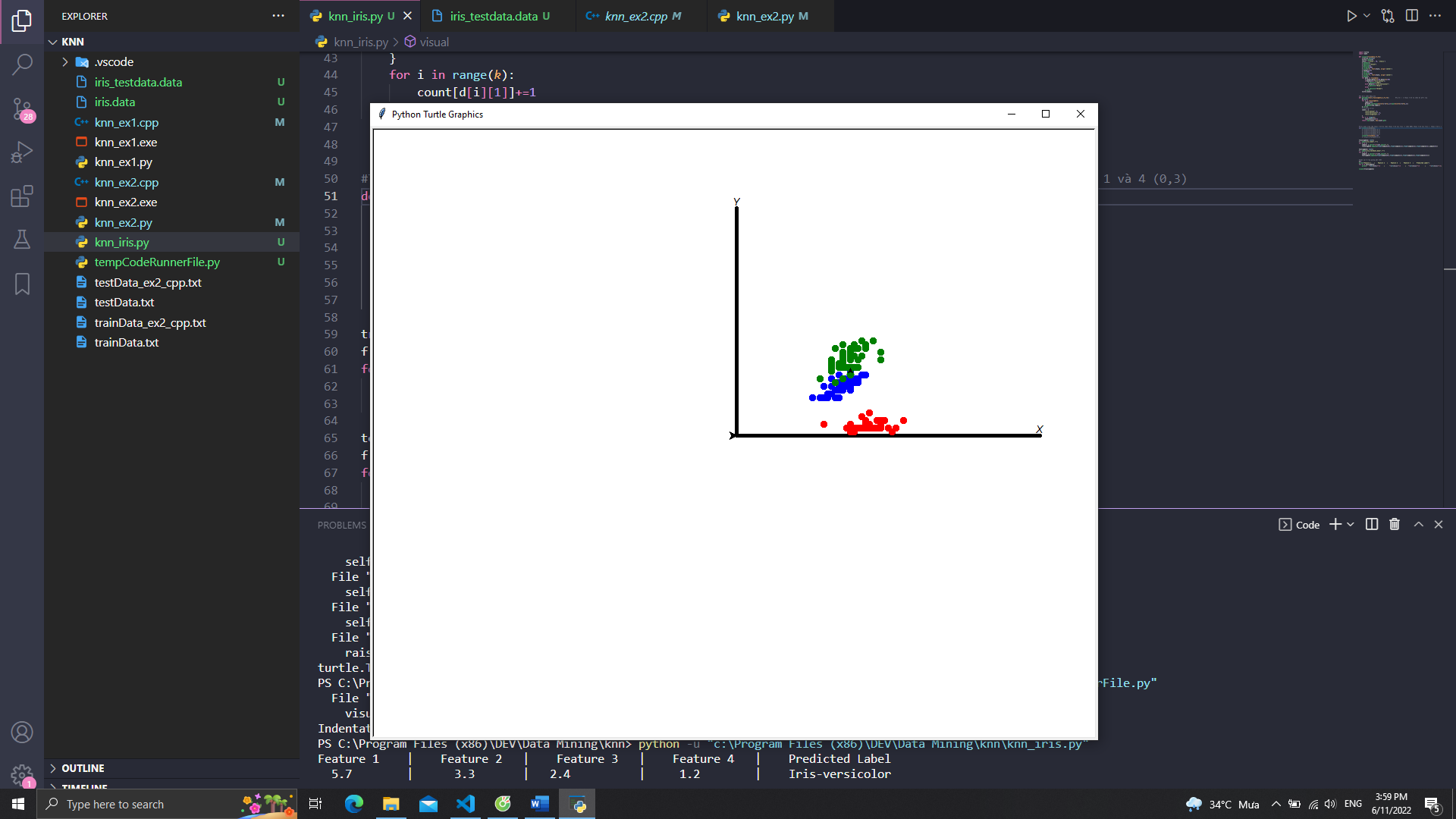
Hình 2. 2Trực quan 2 thuộc tính 1 và 3

**

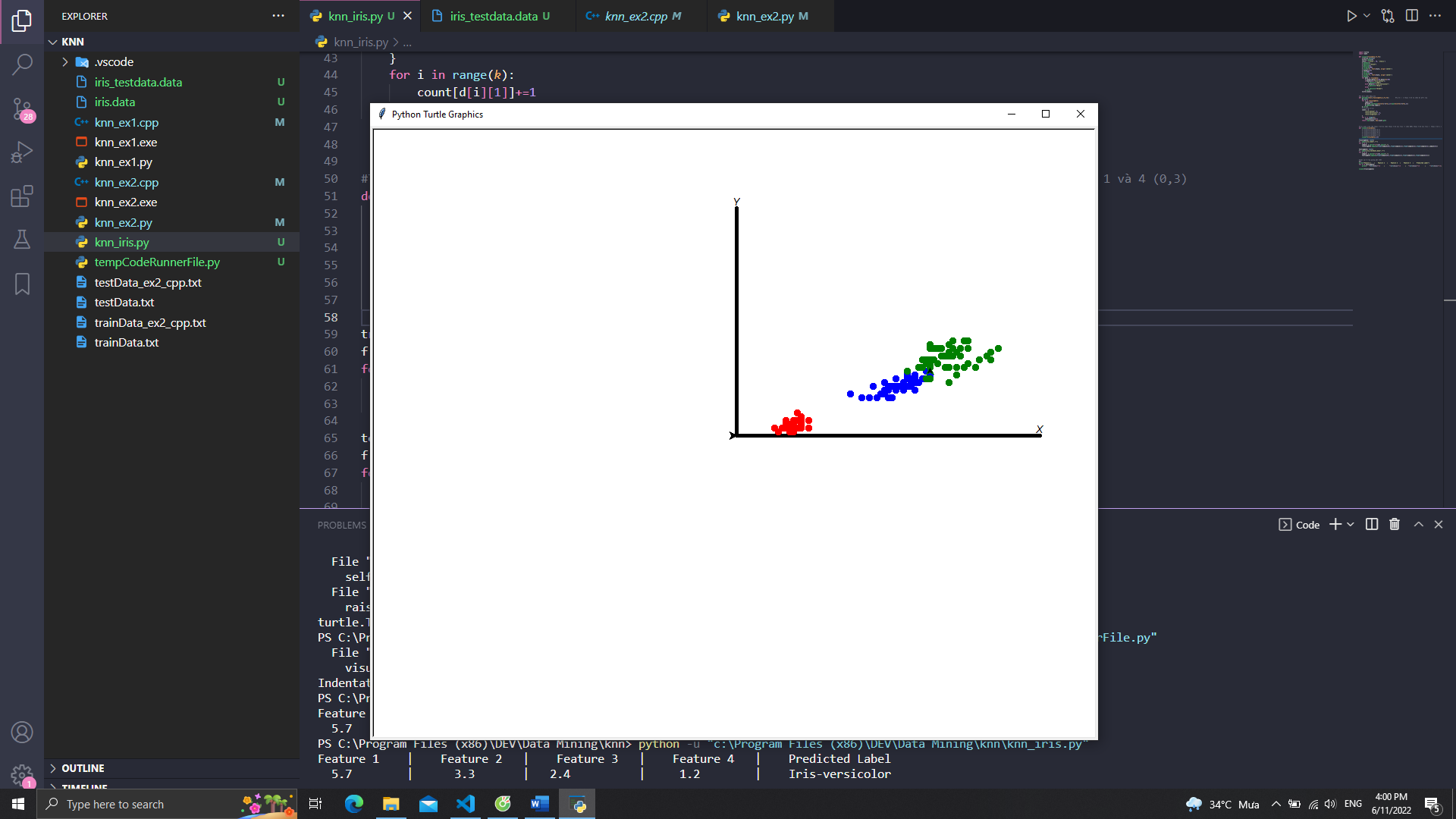
Hình 2. 3Trực quan 2 thuộc tính 1 và 4



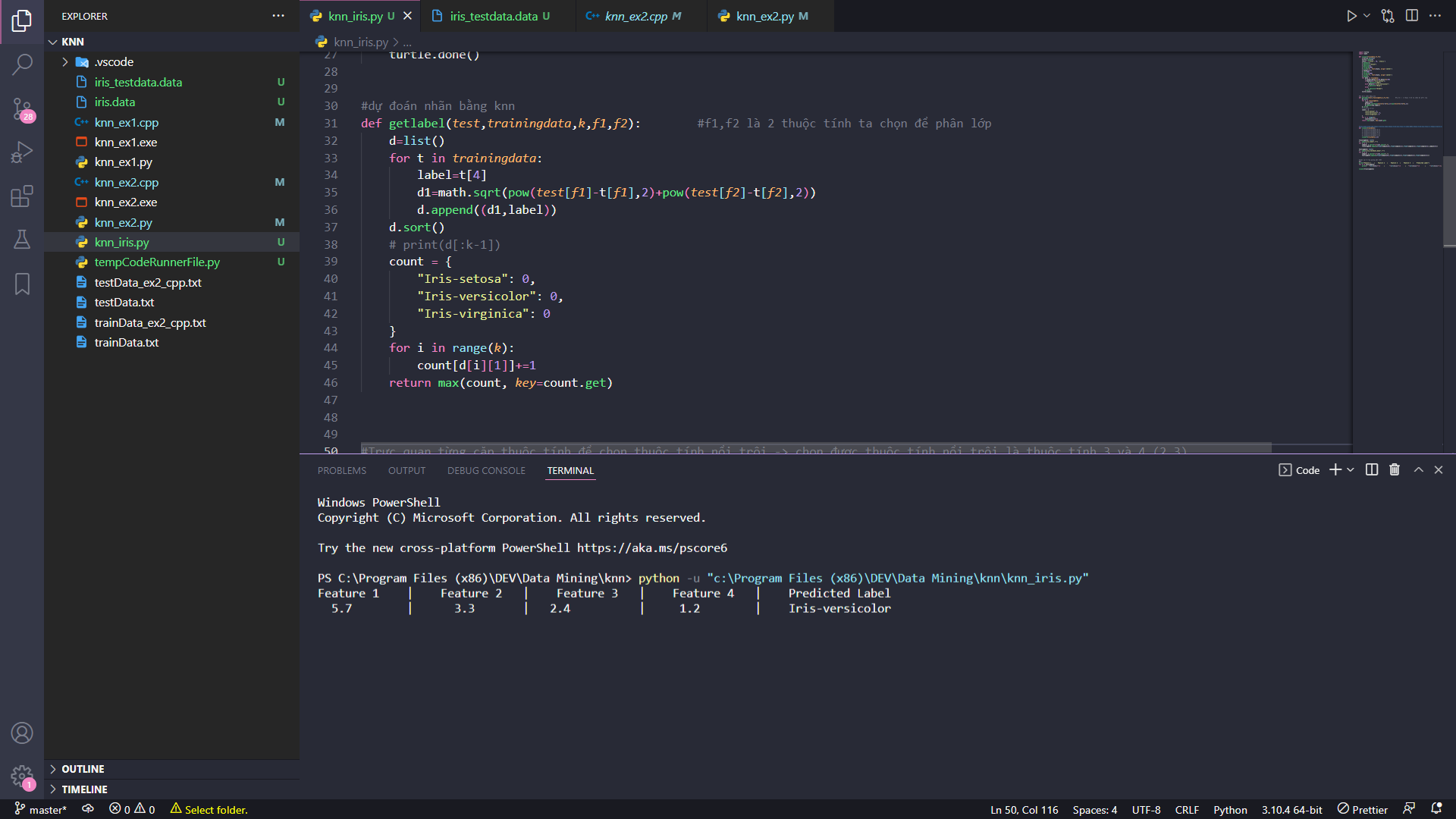
Hình 2. 4 Trực quan 2 thuộc tính 2 và 3



Hình 2. 5 Trực quan 2 thuộc tính 2 và 4

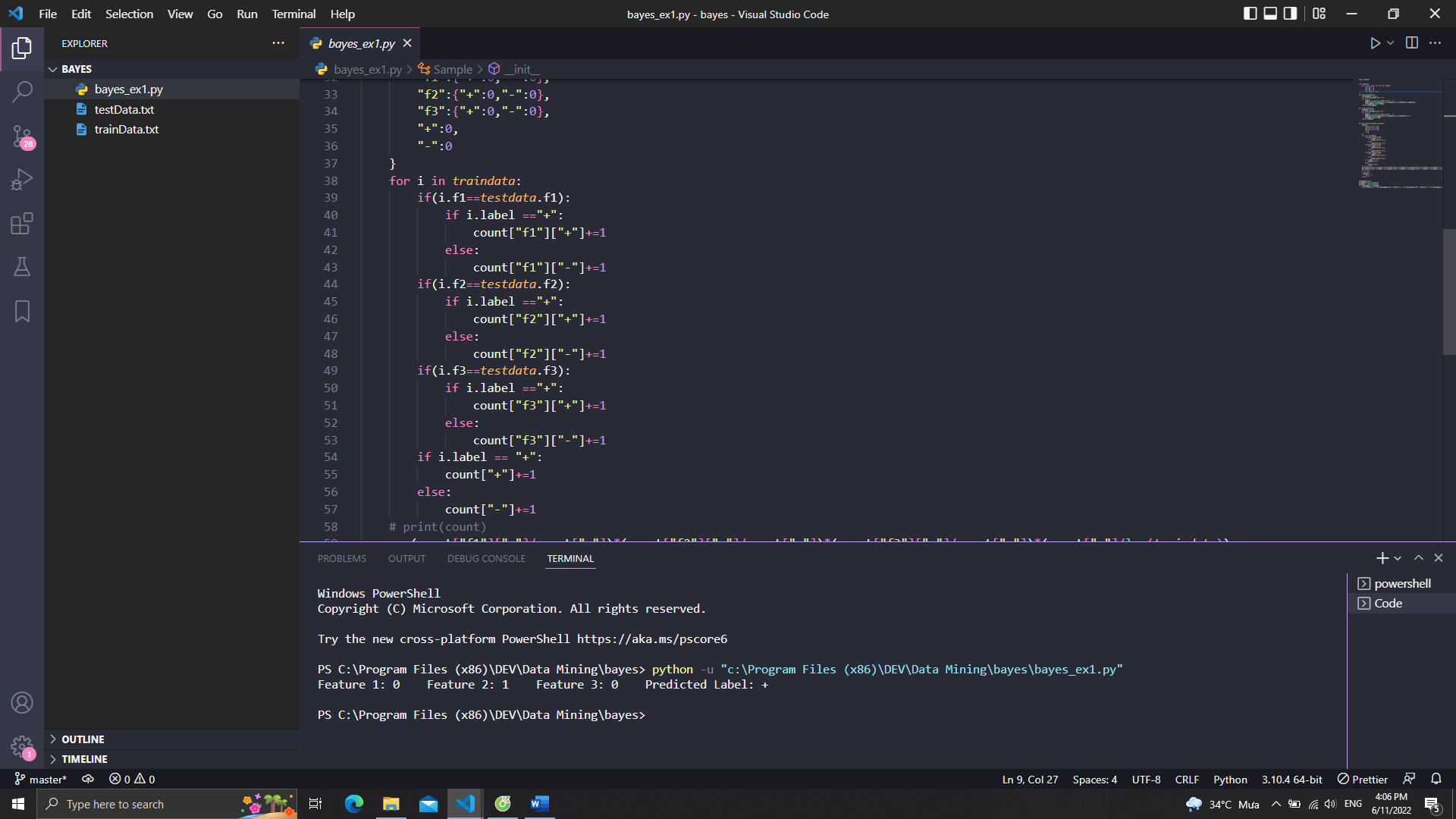


Hình 2. 6 Trực quan 2 thuộc tính 3 và 4



Hình 2. 7 Source Code Knn cho tập dữ liệu Iris

# Table Description automatically generatedEx4: Using Naïve Bayes:



Hình 2. 8 Source Code Bayes

**Ex5: Using Naïve Bayes**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Outlook** | **Temp** | **Humidity** | **Windy** | **Play** |
| 1 | Sunny | Hot | High | False | No |
| 2 | Sunny | Hot | High | True | No |
| 3 | Overcast | Hot | High | False | Yes |
| 4 | Rainy | Mild | High | False | Yes |
| 5 | Rainy | Cool | Normal | False | Yes |
| 6 | Rainy | Cool | Normal | True | No |
| 7 | Overcast | Cool | Normal | True | Yes |
| 8 | Sunny | Mild | High | False | No |
| 9 | Sunny | Cool | Normal | False | Yes |
| 10 | Rainy | Mild | Normal | False | Yes |
| 11 | Sunny | Mild | Normal | True | Yes |
| 12 | Overcast | Mild | High | True | Yes |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Outlook** | **Temp** | **Humidity** | **Windy** | **Play** |
| 13 | Overcast | Hot | Normal | False | Yes |
| 14 | Rainy | Mild | High | True | No |
| **15** | **Sunny** | **Cool** | **High** | **True** | **???** |

Hint:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Outlook** | | | **Temp** | | | **Humidity** | | | **Windy** | | | **Play** | |
|  | Yes | No |  | Yes | No |  | Yes | No |  | Yes | No | Yes | No |
| Sunny | 2 | 3 | Hot | 2 | 2 | High | 3 | 4 | False | 6 | 2 | 9 | 5 |
| Overcast | 4 | 0 | Mild | 4 | 2 | Normal | 6 | 1 | True | 3 | 3 |  |  |
| Rainy | 3 | 2 | Cool | 3 | 1 |  |  |  |  |  |  |  |  |
| **Sunny** | **2/9** | **3/5** | Hot | **2/9** | **2/5** | **High** | **3/9** | **4/5** | **False** | **6/9** | **2/5** | **9/14** | **5/14** |
| **Overcast** | **4/9** | **0/5** | Mild | **4/9** | **2/5** | **Normal** | **6/9** | **1/5** | **True** | **3/9** | **3/5** |  |  |
| **Rainy** | **3/9** | **2/5** | Cool | **3/9** | **1/5** |  |  |  |  |  |  |  |  |

Kết quả tính tay:

P(X|Play=Yes) = P(Outlook = Sunny | Play = yes)

× P(Temp = Cool | Play = yes)

× P(Humidity = High | Play = yes)

× P(Windy = High | Play = yes)

= 2/9 × 3/9 × 3/9 × 3/9= 0.008230452675

Tương tự:

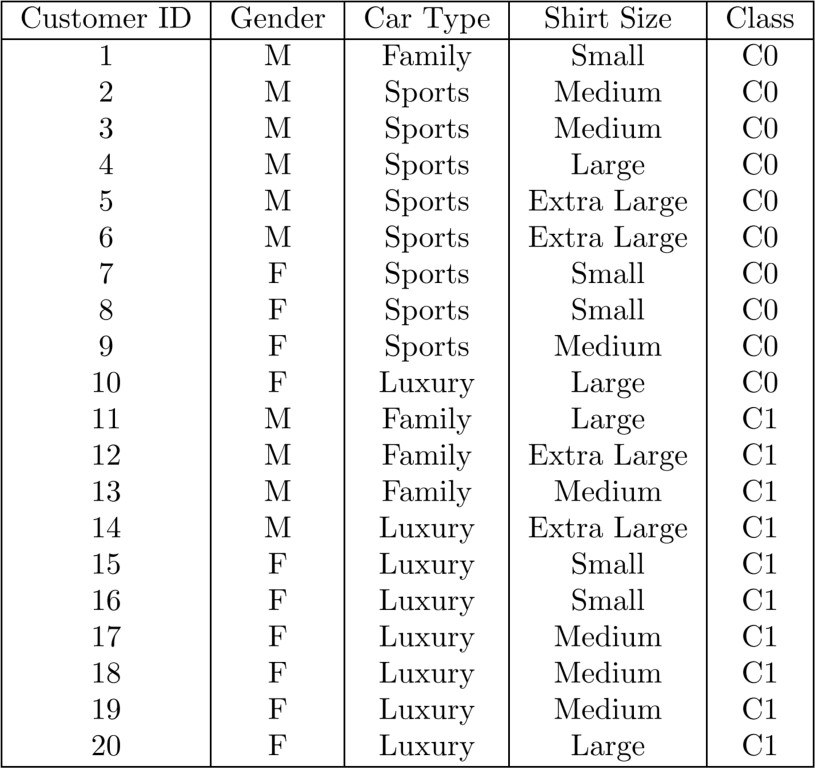
P(X| Play = No) = 3/5 × 1/5 × 4/5 × 3/5= 0.0576

Để tìm ra nhãn(class) ta tính:

P(X| Play=Yes)P(Play = yes) = 0.008230452675× 9/14 = 0.005291005291

P(X| Play = No)P(Play = no) = 0.0576 × 5/14 = 0.2057142857

Vì Vậy, thuật toán Bayes sẽ dự đoán Play=No đối với mẫu X.

**Ex7. Decision tree**

Consider the training examples shown in Table for a binary classification problem

1. Compute the Gini index for the overall collection of training examples.
2. Compute the Gini index for the Customer ID attribute.
3. Compute the Gini index for the Gender attribute.
4. Compute the Gini index for the Car type attribute using multiway split.
5. Compute the Gini index for the Shirt Size using multiway split.
6. Which attribute is better, Gender, Car Type or Shirt size.

Tính chỉ số Gini

Gini(S)= 1 -((∑(p\_i )^2)┬(i=1))┴n

vì bộ traning này có 20 bản ghi và hai lớp có thể có tần số tương đối và (1-p), tương ứng. Trong trường hợp này, C0 và C1 có cùng tần số tương đối (p = 1 – p = 1/2):

Gini = 1 – (p)¬2 - ( 1 – p)2 = 2p (1 - p)=2p2

= 2 × 1/4= 1/2 = 0.5

Tính Gin index cho thuộc tính Customer ID

Gini(S)= -∑\_(i=1)^20▒0 = 0

Vì mỗi bản ghi của thuộc tính này không có tính lặp dữ liệu nên mục gin sẽ là 0

Tính Gin index cho thuộc tính Gender:

P(M) = 10/20 = 0.5 P(F) = 10/20 = 0.5

P(M|C0) = 6/10 = 0.6

P(M|C1) = 4/10 = 0.4

Gini(M) = 1 – ((0.6)2 + (0.4)2) = 0.48

P(F|C0) = 4/10 = 0.4

P(F|C1) = 6/10 = 0.6

Gini(F) = 1 – ((0.4)2 + (0.6)2) = 0.48

Gini(Gender) = 0.5 × 0.48 + 0.5 × 0.48 = 0.48

Tính Gin index cho thuộc tính Car type

P(family) = 4/20 = 0.2; P(luxury) = 8/20 = 0.4; P(sports) = 8/20 = 0.4

P(family|C0) = 1/4 = 0.25

P(family|C1) = 3/4 = 0.75

+ Gini(family) = 1 – ((0.25)2 + (0.75)2) = 0.375

P(luxury|C0) = 1/8 = 0.125

P(luxury|C1) = 7/8 = 0.875

+ Gini(luxury) = 1 – ((0.125)2 + (0.875)2) = 0.219

P(sports|C0) = 8/8 = 1

P(sports|C1) = 0/8 = 0

+ Gini(sports) = 1 – ((1)2 + (0)2) = 0

Gini(Car type) = 0.2 × 0.375 + 0.8 × 0.218 + 0.8 0 = 0.163

Tính Gin index cho thuộc tính Shirt size

P(small) = 5/20 = 0.25 ; P(medium) = 7/20 = 0.35;

P(large) = 4/20 = 0.2 ; P(extra large) = 4/20 = 0.2

P(small|C0) = 3/5 = 0.6

P(small|C1) = 2/5 = 0.4

+ Gini(small) = 1 – ((0.6)2 + (0.4)2) = 0.48

P(medium|C0) = 3/7 = 0.429

P(medium|C1) = 4/7 = 0.571

+ Gini(medium) = 1 – ((0.429)2 + (0.571)2) = 0.49

P(large|C0) = 2/4 = 0.5

P(large|C1) = 2/4 = 0.5

+ Gini(large) = 1 – ((0.5)2 + (0.5)2) = 0.5

P(extra large|C0) = 2/4 = 0.5

P(extra large|C1) = 2/4 = 0.5

+ Gini(extra large) = 1 – ((0.5)2 + (0.5)2) = 0.5

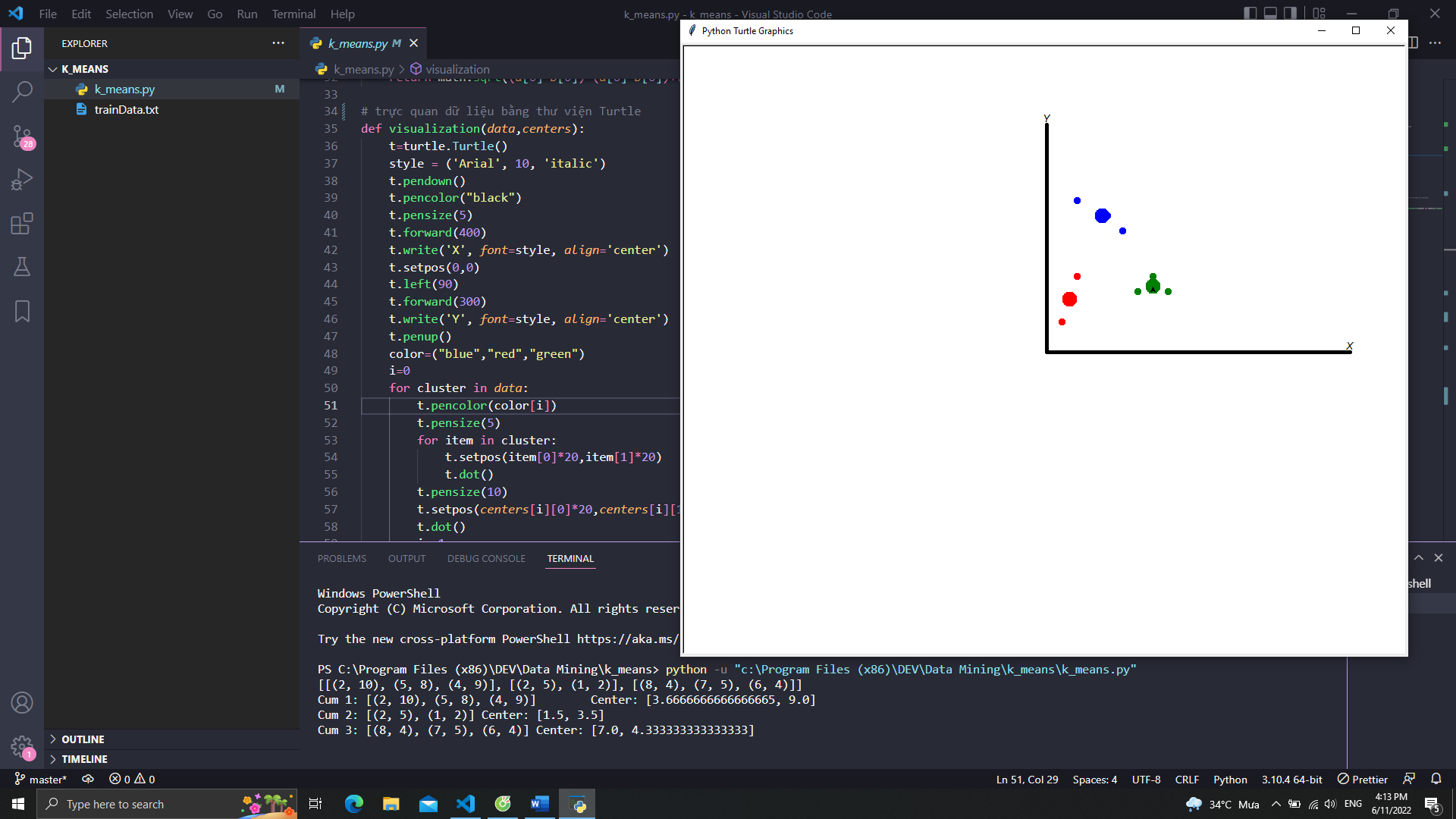
 Gini(Shirt size)= 0.25×0.48 + 0.35×0.49 + 0.2×0.5 + 0.2 ×0.5

= 0.4915

**Ex 7: K-means**

Use the k-means algorithm and Euclidean distance to cluster the following 8 samples with k = 3 clusters: A1=(2,10), A2=(2,5), A3=(8,4), A4=(5,8), A5=(7,5),

A6=(6,4), A7=(1,2), A8=(4,9). Visulization after clustering.



Hình 2. 9 Source Code K-means