**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN HỆ THỐNG THÔNG TIN**

**--------------------**



**NGUYỄN THỊ HIỀN: 15110201**

**ÔNG THỊ DIỄM PHÚC: 15110280**

Đề Tài:

**PHÂN LỚP DỰA TRÊN PHƯƠNG PHÁP PRM**

**BÁO CÁO KHÓA LUẬN TỐT NGHIỆP**

**GIÁO VIÊN HƯỚNG DẪN**

**TS. NGUYỄN THÀNH SƠN**

**KHOÁ 2015 - 2019**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN HỆ THỐNG THÔNG TIN**

**--------------------**



**NGUYỄN THỊ HIỀN: 15110201**

**ÔNG THỊ DIỄM PHÚC: 15110290**

Đề Tài:

**PHÂN LỚP DỰA TRÊN PHƯƠNG PHÁP PRM**

**BÁO CÁO KHÓA LUẬN TỐT NGHIỆP**

**GIÁO VIÊN HƯỚNG DẪN**

**TS. NGUYỄN THÀNH SƠN**

**KHOÁ 2015 - 2019**

**ĐH SƯ PHẠM KỸ THUẬT TP.HCM CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**KHOA CÔNG NGHỆ THÔNG TIN Độc lập – Tự do – Hạnh phúc**

**------------ ------------**

**PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN**

Họ tên sinh viên thực hiện 1: Nguyễn Thị Hiền MSSV: 15110201

Họ tên sinh viên thực hiện 2: Ông Thị Diễm Phúc MSSV: 15110280

Ngành: Công Nghệ Thông Tin

Tên đề tài: Phân lớp dựa trên luật dự đoán PRM.

Họ và tên Giảng viên phản biện:

**NHẬN XÉT**

1. Về nội dung đề tài và khối lượng công việc:

2. Ưu điểm:

3. Khuyết điểm:

4. Đề nghị cho bảo vệ hay không?

5. Đánh giá loại:

6. Điểm:

TP. Hồ Chí Minh, ngày … tháng … năm 2019

Giảng viên phản biện

**LỜI CÁM ƠN**

Lời đầu tiên, nhóm chúng em xin chân thành cảm ơn khoa Công Nghệ Thông Tin, bộ môn Hệ Thống Thông Tin, trường Đại Học Sư Phạm Kỹ Thuật TP.HCM đã tạo điều kiện thuận lợi cho chúng em thực hiện đề tài này.

Đồng thời, nhóm chúng em cũng xin gửi lời cảm ơn chân thành nhất đến thầy Nguyễn Thành Sơn, người đã tận tình chỉ bảo và hướng dẫn nhóm chúng em thực hiện đề tài này.

Bên cạnh đó, nhóm chúng em xin gửi lời cảm ơn chân thành đến các thầy cô giảng viên trường Đại Học Sư Phạm Kỹ Thuật nói chung, cũng như các thầy cô giảng viên khoa Công Nghệ Thông Tin nói riêng, những người đã tạo điều kiện cho chúng em tích lũy được những kiến thức quý báu trong những năm học qua.

Nhóm chúng em cũng không quên gửi lời biết ơn sâu sắc đến cha mẹ, gia đình, bạn bè, những người đã bên cạnh, ủng hộ, chăm sóc, động viên chúng em trong suốt thời gian học tập và nghiên cứu.

Mặc dù đã cố gắng hoàn thành đề tài khóa luận đúng yêu cầu, nhưng do thời gian hạn hẹp, khả năng còn hạn chế nên khó tránh khỏi việc thiếu sót trong bài báo cáo. Chúng em rất mong nhận được sự thông cảm, góp ý cũng như đóng góp của quý thầy cô và các bạn để đề tài hoàn thiện hơn, để có thể phát triển hơn phục vụ cho tương lai.

Chúng em chân thành cảm ơn.

**ĐH SƯ PHẠM KỸ THUẬT TP.HCM CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**KHOA CÔNG NGHỆ THÔNG TIN Độc lập – Tự do – Hạnh phúc**

**------------ ------------**

**ĐỀ CƯƠNG KHOÁ LUẬN TỐT NGHIỆP**

Họ tên sinh viên thực hiện 1: Nguyễn Thị Hiền MSSV: 15110201

Họ tên sinh viên thực hiện 2: Ông Thị Diễm Phúc MSSV: 15110280

Thời gian làm khóa luận: Từ 17/02/2019 Đến 19/06/2019

Chuyên ngành: Hệ Thống Thông Tin.

Tên đề tài: Phân lớp dựa trên phương pháp PRM.

Giảng viên hướng dẫn: TS. Nguyễn Thành Sơn

**Nhiệm vụ của khoá luận**

1. Tìm hiểu tổng quan phương pháp PRM.

2. Tìm hiểu phương pháp cải tiến những khuyết điểm của thuật toán PRM: xử lý thuật toán khi dữ liệu đầu vào có missing values.

3. Suy nghĩ và đề xuất một ý tưởng để hoàn thiện việc cải tiến thuật toán PRM

4. Tìm hiểu tổng quan về Python và các công cụ.

5. Suy nghĩ và đề xuất ý tưởng tiếp tục cải tiến PRM, chứng minh bằng thực nghiệm.

**KẾ HOẠCH THỰC HIỆN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thời gian** | **Công Việc** | **Kết quả** | **Ghi chú** |
| **1** | Từ 17/02/2019  Đến 24/02/2019 | Đọc, tìm hiểu tài liệu về phương pháp PRM, hướng cải tiến phương pháp | 95% |  |
| **2** | Từ 25/02/2019  Đến 03/03/2019 | Triển khai các cách để áp dụng các phương pháp tìm được vào chương trình | 90% |  |
| **3** | Từ 04/003/2019  Đến 11/03/2019 | Chỉnh sửa giao diện thân thiện người dùng. | 90% |  |
| **4** | Từ 12/03/2019  Đến 10/04/2019 | Tìm hiểu thuật toán PRM  Nhận xét ưu, nhược điểm | 90% |  |
| **5** |
| **6** |
| **7** |
| **8** | Từ 11/04/2019  Đến 25/04/2019 | Suy nghĩ, lên ý tưởng và đề xuất phương hướng cải tiến | 85% |  |
| **9** |
| **10** | Từ 26/04/2019  Đến 17/05/2019 | Kiểm tra chương trình, sử dụng các phương pháp cải tiến trước đó đã tìm hiểu vào chương trình: xử lý được nguồn dữ liệu lớn nhưng lại tiết kiệm thời gian, xử lý được dữ liệu từ nhều dạng, … | 90% |  |
| **11** |
| **12** |
| **13** | Từ 18/05/2019  Đến 25/05/2019 | Tổng hợp viết báo cáo | 90% |  |
| **14** | Từ 26/05/2019  Đến 03/06/2019 | Chỉnh sửa lại một số nội dung trong báo cáo theo yêu cầu GVHD. | 95% |  |
| **15** | Từ 04/06/2019  Đến 11/06/2019 | Hoàn chỉnh báo cáo và chỉnh sửa code. | 90% |  |

Ý kiến giảng viên hướng dẫn TP. Hồ Chí Minh, ngày … tháng … năm 2019

(Ký và ghi rõ họ tên) Người viết đề cương

MỤC LỤC

[**1.1.** **TÍNH CẤP THIẾT CỦA ĐỀ TÀI** 9](#_Toc13954135)

[**1.2.** **MỤC ĐÍCH CỦA ĐỀ TÀI** 9](#_Toc13954136)

[**1.3.** **KẾT QUẢ DỰ KIẾN ĐẠT ĐƯỢC** 10](#_Toc13954137)

[**Phần 2: NỘI DUNG** 11](#_Toc13954138)

[**Chương 1: KIẾN THỨC TỔNG QUÁT** 11](#_Toc13954139)

[1.1. PYTHON 11](#_Toc13954140)

[1.2. ƯU ĐIỂM NỔI BẬT CỦA PYTHON 11](#_Toc13954141)

[1.3. CÁC BƯỚC CÀI ĐẶT PYTHON 12](#_Toc13954142)

[1.4. CÁC CÔNG CỤ LẬP TRÌNH PYTHON CƠ BẢN 13](#_Toc13954143)

[1.5. LÝ DO CHỌN PYCHARM 17](#_Toc13954144)

[1.6. CÔNG CỤ LẬP TRÌNH PYCHARM 18](#_Toc13954145)

[1.6.1. Tính năng nổi bật của PyCharm 18](#_Toc13954146)

[1.6.2. Các bước cài đặt PyCharm 18](#_Toc13954147)

[1.6.3. Hướng dẫn tạo project Python trên phần mềm Pycharm 21](#_Toc13954148)

[**Chương 2: KHAI PHÁ LUẬT KẾT HỢP** 25](#_Toc13954149)

[2.1. BÀI TOÁN KHAI PHÁ LUẬT KẾT HỢP 25](#_Toc13954150)

[2.2. MỘT SỐ KHÁI NIỆM CƠ BẢN 25](#_Toc13954151)

[2.2.1. Literal 26](#_Toc13954152)

[2.2.2. Định nghĩa luật kết hợp 26](#_Toc13954153)

[2.2. CÁC THUẬT TOÁN CƠ BẢN 26](#_Toc13954154)

[2.2.1. Thuật toán Apriori 26](#_Toc13954155)

[2.2.2. Thuật toán Apriori-TID 27](#_Toc13954156)

[2.2.3. Logistics Regression 27](#_Toc13954157)

[2.2.4. Support Vector Machine (SVM) cho bài toán phân lớp quan điểm 27](#_Toc13954158)

[**Chương 3: PHÂN LỚP DỰA TRÊN LUẬT KẾT HỢP** 29](#_Toc13954159)

[3.1. TỔNG QUAN VỀ PHÂN LỚP DỰA TRÊN LUẬT KẾT HỢP 29](#_Toc13954160)

[3.2. CÁC CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN 29](#_Toc13954161)

[3.2.1. FOIL (first order inductive leaner) 29](#_Toc13954162)

[**3.2.2. Thuật toán phân lớp kết hợp sử dụng tối ưu hóa đàn kiến (ACO)** 30](#_Toc13954163)

[3.2.3. Cải tiến thuật toán CAR (An Efficient Algorithm for Mining ClassAssociation Rules with Interestingness Measures) 31](#_Toc13954164)

[3.2.4. Phân lớp nhanh dựa trên luật kết hợp (Fast Classification Based on Association Rules Algorithm) 31](#_Toc13954165)

[3.2.5. Khai thác quy tắc kết hợp (ARM) 32](#_Toc13954166)

[3.3. THUẬT TOÁN PRM (PREDICTIVE RULE MINING) 32](#_Toc13954167)

[3.3.1. Tổng quan 32](#_Toc13954168)

[3.3.2. Giới thiệu tập Positive and Negative. 33](#_Toc13954169)

[3.3.3. Thuật toán 33](#_Toc13954170)

[3.3.4. Ví dụ 34](#_Toc13954171)

[3.5. PHÂN LỚP DỰA TRÊN LUẬT KẾT HỢP ĐÃ TÌM THẤY 45](#_Toc13954172)

[**Chương 4: ĐỀ XUẤT Ý TƯỞNG CẢI TIẾN.** 47](#_Toc13954173)

[4.1. TRÌNH BÀY Ý TƯỞNG 47](#_Toc13954174)

[4.2. THỰC HIỆN Ý TƯỞNG 47](#_Toc13954175)

[**Chương 5: THỰC NGHIỆM** 50](#_Toc13954176)

[5.1. MÔI TRƯỜNG THỰC NGHIỆM 50](#_Toc13954177)

[5.2. DỮ LIỆU THỰC NGHIỆM 51](#_Toc13954178)

[**5.2.1. Bệnh nhân tiểu đường ấn độ (Pima Indians)** 51](#_Toc13954179)

[**5.2.2.** **Hồ sơ bệnh nhân gan Ấn Độ (Indian Liver Patient Records)** 52](#_Toc13954180)

[**5.2.3.** **Phân loại nấm (Mushroom Classification).** 53](#_Toc13954181)

[5.2.4. Phân loại bệnh tim 54](#_Toc13954182)

[5.2.5. Xếp loại sinh viên 56](#_Toc13954183)

[5.3. QUY TRÌNH THỰC NGHIỆM VÀ TIÊU CHÍ ĐÁNH GIÁ 58](#_Toc13954184)

[5.3.1. Tạo tập dữ liệu có missing value 58](#_Toc13954185)

[5.3.2. Tiêu chí đánh giá. 58](#_Toc13954186)

[**5.3.3. Các bước thực nghiệm.** 58](#_Toc13954187)

[5.4. KẾT QUẢ CHẠY THỰC NGHIỆM 59](#_Toc13954188)

[5.4.1. Dataset: bệnh nhân tiểu đường (Pima Indians) 59](#_Toc13954189)

[5.4.2. Dataset bệnh gan (Indian Liver Patient Records) 62](#_Toc13954190)

[5.4.3. Dataset phân loại nấm (Mushroom Classification) 65](#_Toc13954191)

[5.4.4. Dataset phân loại bệnh tim (Heart) 68](#_Toc13954192)

[**Phần 3: KẾT LUẬN** 70](#_Toc13954193)

[**3.1. KẾT QUẢ SAU KHI HOÀN THÀNH BÁO CÁO** 70](#_Toc13954194)

[**3.2. KHÓ KHĂN GẶP PHẢI** 70](#_Toc13954195)

[**3.3. ƯU ĐIỂM VÀ NHƯỢC ĐIỂM** 70](#_Toc13954196)

[**3.3. HƯỚNG PHÁT TRIỂN ĐỀ TÀI TRONG TƯƠNG LAI** 71](#_Toc13954197)

**Phần 1: MỞ ĐẦU**

## **TÍNH CẤP THIẾT CỦA ĐỀ TÀI**

Với sự bùng nổ và phát triển của công nghệ thông tin đã mang lại nhiều hiệu quả đối với khoa học cũng như các hoạt động thực tế, trong đó khai phá dữ liệu là một lĩnh vực mang lại hiệu quả thiết thực cho con người. Khai phá dữ liệu đã giúp người sử dụng thu được những tri thức hữu ích từ những cơ sở dữ liệu hoặc các kho dữ liệu khổng lồ khác.

Cơ sở dữ liệu trong các đơn vị, tổ chức kinh doanh, quản lý khoa học chứa đựng nhiều thông tin tiềm ẩn, phong phú và đa dạng, đòi hỏi phải có những phương pháp nhanh, phù hợp, chính xác, hiệu quả để lấy được những thông tin bổ ích. Những “*tri thức*” chiết suất từ nguồn cơ sở dữ liệu trên sẽ là nguồn thông tin hỗ trợ cho lãnh đạo trong việc lên kế hoạch hoạt động hoặc trong việc ra quyết định sản xuất kinh doanh. Tiến hành công việc như vậy chính là thực hiện quá trình phát hiện tri thức trong cơ sở dữ liệu (Knowledge Discovery in Database) mà trong đó kỹ thuật khai phá dữ liệu (Data Mining) cho phép phát hiện những tri thức tiềm ẩn. Để lấy được thông tin mang tính tri thức trong khối dữ liệu khổng lồ, cần thiết phải phát triển các kỹ thuật có khả năng tích hợp các dữ liệu từ các hệ thống giao dịch khác nhau, chuyển chúng thành một tập hợp các cơ sở dữ liệu ổn định có chất lượng. Các kỹ thuật như vậy được gọi là kỹ thuật tạo kho dữ liệu và môi trường các dữ liệu nhận được khi áp dụng các kỹ thuật tạo kho dữ liệu nói trên được gọi là kho dữ liệu (Data Warehouse).

Một trong các nội dung cơ bản nhất trong khai phá dữ liệu và rất phổ biến là phát hiện các luật kết hợp. Phương pháp này nhằm tìm ra các thuộc tính thường xuất hiện đồng thời trong cơ sở dữ liệu và rút ra các luật về ảnh hưởng của một thuộc tính dẫn đến sự xuất hiện của một thuộc tính khác như thế nào.

Ứng dụng khai phá dữ liệu đã mang lại những lợi ích to lớn trong việc tổng hợp và cung cấp những thông tin trong các nguồn cơ sở dữ liệu lớn.

* 1. **MỤC ĐÍCH CỦA ĐỀ TÀI**

Sau quá trình nghiên cứu và tìm hiểu thuật toán CPAR đã được cải tiến, nhóm đã tìm ra được những khuyết điểm và sự bất tiện cùng thiếu chính xác của việc cải tiến thuật toán theo phương pháp cũ. Nên chúng em đã quyết định đề xuất ý tưởng và cải tiến thuật toán theo một phương pháp khác

* 1. **KẾT QUẢ DỰ KIẾN ĐẠT ĐƯỢC**

Kết quả mong muốn ban đầu của nhóm là chứng minh được việc cải tiến thuật toán theo phương pháp mới sẽ làm tăng độ chính xác hơn so với phương pháp cũ.

Nội dung luận văn được trình bày trong 5 chương và phần kết luận:

* Chương 1: Tổng quan về khai phá dữ liệu: Giới thiệu tổng quan về quá trình khai phá dữ liệu, kho dữ liệu và khai phá dữ liệu; kiến trúc của một hệ thống khai phá dữ liệu; Nhiệm vụ chính và các phương pháp khai phá dữ liệu.
* Chương 2: Tổng quan về Python và công cụ Pycharm
* Chương 3: Thuật Toán PRM
* Chương 4: Trình bày ý tưởng cải tiến thuật toán mới
* Chương 5: Chứng minh bằng thực nghiệm

# **Phần 2: NỘI DUNG**

## **Chương 1: KIẾN THỨC TỔNG QUÁT**

### 1.1. PYTHON

Python là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) bậc cao cho các mục đích lập trình đa năng, do [Guido van Rossum](https://vi.wikipedia.org/w/index.php?title=Guido_van_Rossum&action=edit&redlink=1) tạo ra và lần đầu ra mắt vào năm [1991](https://vi.wikipedia.org/wiki/1990). Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Nhìn nhận chung thì Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu.

Nhà phát triển: Python Software Foundation.

### 1.2. ƯU ĐIỂM NỔI BẬT CỦA PYTHON

- Rất mạnh trong việc xử lý các loại dữ liệu chuỗi, tập hợp. Thích hợp với ứng dụng phân tách, chuyển đổi, phân tích dữ liệu: big data - data mining.

- Dễ học - dễ làm - dễ cài đặt.

- Chạy trên đa nền tảng: MacOSX, Windows, Linux.

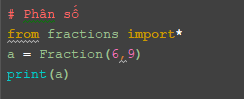
- Lập trình gần như mọi thứ: Web (Django, Tornado), Game (pygame , kivy, piglet), lập trình nhúng, phân tích dữ liệu.

- Thư viện có sẵn nhiều. Ví dụ thư viện data mining Scikit-learn , Pandas ...

- Tại thị trường tuyển dụng tại Việt Nam: Python là hàng độc so với kỹ năng lập trình PHP, C#, Java.

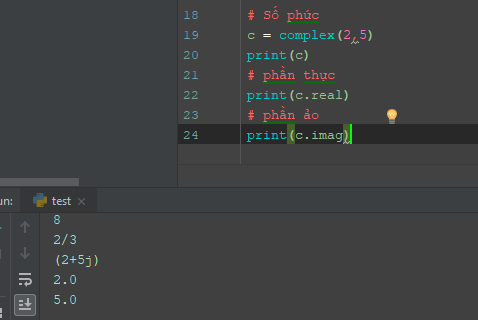
- Một vài điểm nổi trội của python so với C, C++, …: Hỗ trợ cả kiểu dữ liệu phân số và số phức.

+ Khai báo kiểu phân số: import nội dung thư viện fractions theo đoạn code sau:



**Hình 1.1 Khai báo kiểu phân số**

+ Khai báo kiểu số thực theo đoạn code sau:



**Hình 1.2. Khai báo kiểu số thực**

Tự gán kiểu dữ liệu phù hợp với dữ liệu gán, ví dụ:

- a = 3 (tự động hiểu a là kiểu nguyên).

- b = “Hoa” (tự hiểu kiểu string).

- c =1.9 (tự động hiểu là kiểu float: số thực và lấy xấp xỉ 15 chữ số thập phân).

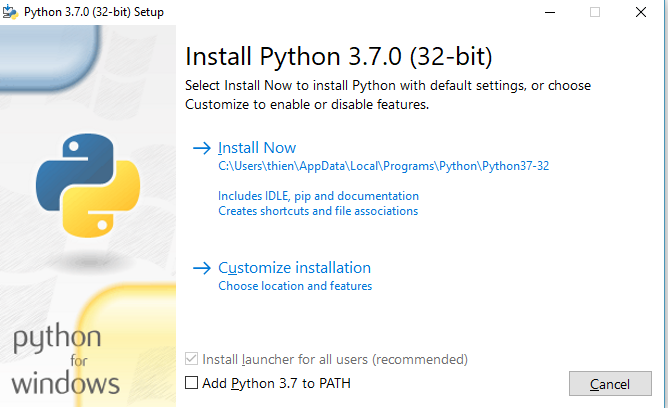
print(type(<biến>)): để xem kiểu.

Không giới hạn miền giá trị của kiểu dữ liệu, ví dụ: kiểu int có thể in một số bất kỳ so với C, C#, …

### 1.3. CÁC BƯỚC CÀI ĐẶT PYTHON

*Phiên bản 3.7.0*

Sau khi tải file cài đặt Python, bấm chọn file đã tải, hộp thoại sau xuất hiện.Chọn Install Now để tiến hành cài đặt Python theo mặc định Chọn Install Now, sau đó bấm Next như sau:



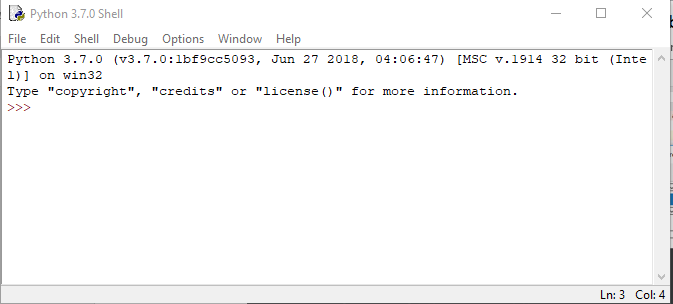
**Hình 1.3. Cài đặt python**

Sau đó chờ quá trình cài đặt hoàn tất.

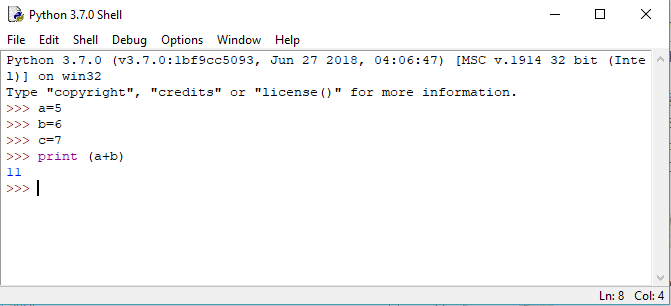
### 1.4. CÁC CÔNG CỤ LẬP TRÌNH PYTHON CƠ BẢN

**Công cụ lập trình chính thống khi cài đặt Python: IDLE**

*Link:* [*https://www.python.org/*](https://www.python.org/)

Sau khi cài đặt thành công Python, vào Start chọn IDLE (Python 3.7). Thì Giao diện Python IDLE sẽ suất hiện như bên dưới.

**Hình 1.4.Giao diện python**

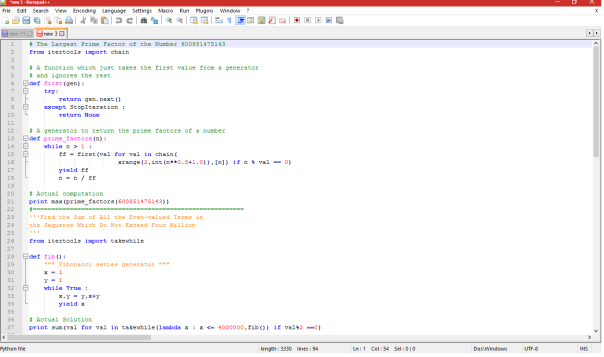
Ta có thể gõ các lệnh trực tiếp vào màn hình này như ví dụ bên dưới.

**Hình 1.5.Ví dụ code trên IDEL**

Hình trên là ví dụ cho việc gán giá trị a, b, c mà không cần khai báo biến.

**Notepad++**

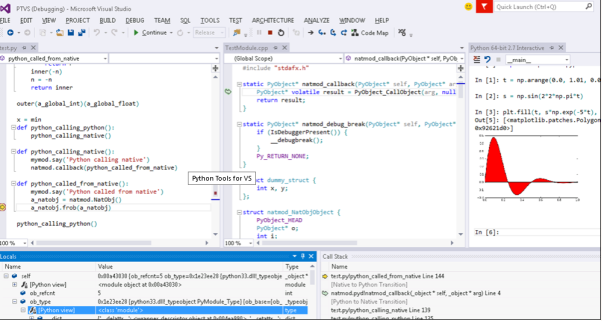
Link: <https://notepad-plus-plus.org/>

Giao diện Notepad++ như bên dưới.

**Hình 1.6.Lập Trình python với Notepad++**

**Python Tools for Visual Studio**

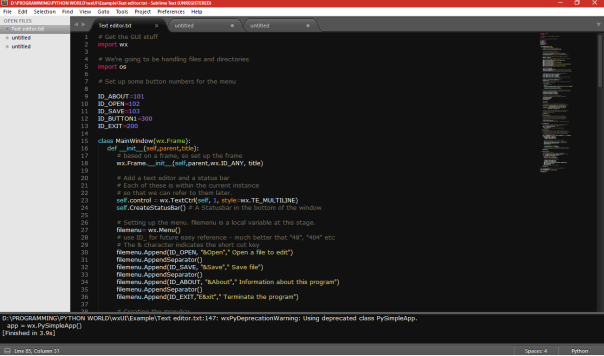
Link: <https://microsoft.github.io/PTVS/>

Giao diện Python Tools for Visual Studio suất hiện như hình bên dưới.  


**Hình 1.7 Lập trình python với Visual Studio**

**Sublime Text**

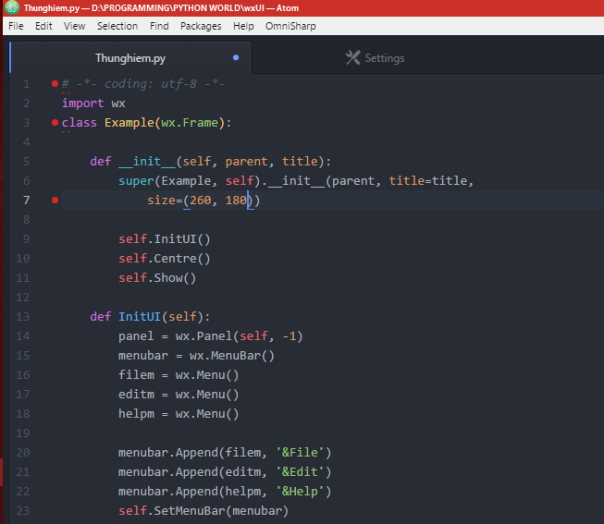
Link: <https://www.sublimetext.com/>

Giao diện Python Tools for Visual Studiosuất hiện như trước.

**Hình 1.8.Lập trình python với Sublime Text**

**Atom**

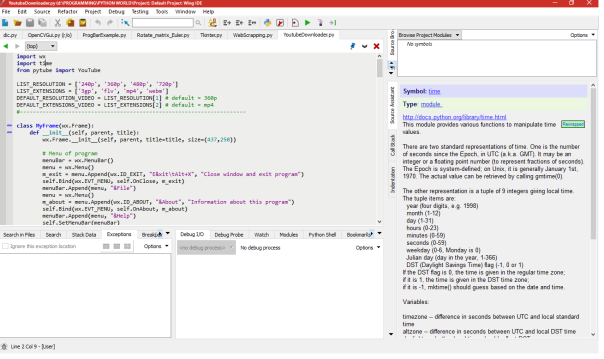
Link: <https://atom.io/>

Giao diện Atom như hình bên dưới.

**Hình 1.9.Lập trình python với Atom**

**Công cụ lập trình wingide**

Link web: <http://wingware.com/downloads/wingide-101>

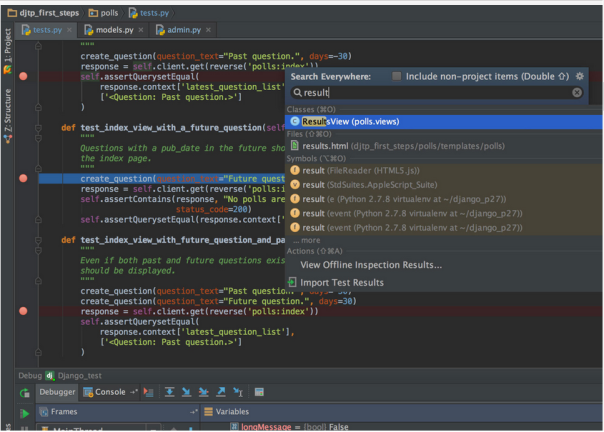
Giao diện wingide có dạng như hình bên dưới.

**Hình 1.10.lập trình python với Wingide**

**Công cụ lập trình PyCharm:**

*Link:* [*https://www.jetbrains.com/pycharm/*](https://www.jetbrains.com/pycharm/)

Giao diện PyCharm suất hiện như hình bên dưới.



**Hình 1.11.Lập trình python với PyCharm**

### 1.5. LÝ DO CHỌN PYCHARM

Chúng em chọn Pycharm vì nó có nhiều khả năng nổi bật như kiểm tra đơn vị tích hợp, kiểm tra mã, kiểm soát phiên bản tích hợp, công cụ tái cấu trúc mã, nhiều công cụ để điều hướng dự án, cũng như các tính năng nổi bật và hoàn thiện tự động mà người lập trình mong đợi ở một IDE.

Ngôn ngữ được hỗ trợ: AngularJS, CSS, Cython, HTML, JavaScript, Node.js, Python, TypeScript, Coffee Script và các ngôn ngữ mẫu.

Ưu điểm:

Tương thích với Windows, Linux và Mac OS.

Đi kèm với Django IDE.

Dễ dàng tích hợp với Git, Mercurial và SVN.

Giao diện tùy biến với mô phỏng VIM.

Trình sửa lỗi JavaScript, Python và Django.

Hỗ trợ Google App Engine.

Nhược điểm: Sau khi sử dụng PyCharm một thời gian thì nhóm nhận thấy có một số lỗi, chẳng hạn như tính năng tự động hoàn thành đôi khi không hoạt động, có thể là một sự bất tiện nhỏ.

Nhược điểm chính của PyCharm là mô hình mở của nó. Nhiều tính năng tiên tiến của PyCharm không có sẵn theo giấy phép nguồn mở. Tuy nhiên, nếu bạn không muốn sử dụng các tính năng nâng cao hơn trong verion nguồn đóng, có các tính năng còn lại có thể khiến PyCharm trở thành lựa chọn trọng lượng nhẹ hơn để chỉnh sửa Python**.**

Quan trọng nhất là trong tất cả các bảng xếp hạng IDE hỗ trợ python thì pycharm luôn có vị trí trong đó và thậm chí là vị trí cao, còn về các IDE hỗ trợ cho editors thì hầu như luôn đứng đầu.

Ví dụ: “13 Best IDE And Editors For Python In 2018” By Deven Rathore

<https://www.dunebook.com/best-python-ide-windows-mac/>

“Top 3 open source Python IDEs" by Jason Baker (Red Hat)

<https://opensource.com/resources/python/ides>

“THE BEST 1 OF 37 OPTIONS”

<https://www.slant.co/topics/366/~best-python-ides-or-editors>

"Top 5 Python IDEs For Data Science" by Paulo Henrique Vasconcellos

<https://www.datacamp.com/community/tutorials/data-science-python-ide>

10 IDE LÝ TƯỞNG CHO LẬP TRÌNH VIÊN

<http://www.ifisolution.com/?p=466>

### 1.6. CÔNG CỤ LẬP TRÌNH PYCHARM

PyCharm là một môi trường phát triển tích hợp (IDE) được sử dụng trong lập trình máy tính, đặc biệt cho ngôn ngữ Python. Nó được phát triển bởi công ty JetBrains của Séc. Nó cung cấp phân tích mã, trình gỡ lỗi đồ họa, trình kiểm tra đơn vị tích hợp, tích hợp với các hệ thống điều khiển phiên bản (VCS), và hỗ trợ phát triển web với Django.

PyCharm là nền tảng đa nền tảng với các phiên bản Windows, macOS và Linux. Phiên bản Cộng đồng được phát hành theo Giấy phép Apache và cũng có phiên bản Chuyên nghiệp được phát hành theo giấy phép sở hữu độc quyền - điều này có các tính năng bổ sung.

#### 1.6.1. Tính năng nổi bật của PyCharm

- Mã hóa hỗ trợ và phân tích, với hoàn thành mã, cú pháp và làm nổi bật lỗi, tích hợp linter và sửa lỗi nhanh.

- Chuyển hướng dự án và mã: các khung nhìn dự án chuyên biệt, khung nhìn cấu trúc tệp và nhảy nhanh giữa các tệp, các lớp, các phương thức và cách sử dụng.

- Tái cấu trúc Python: bao gồm đổi tên, phương thức trích xuất, giới thiệu biến, giới thiệu liên tục, kéo lên, đẩy xuống và những thứ khác.

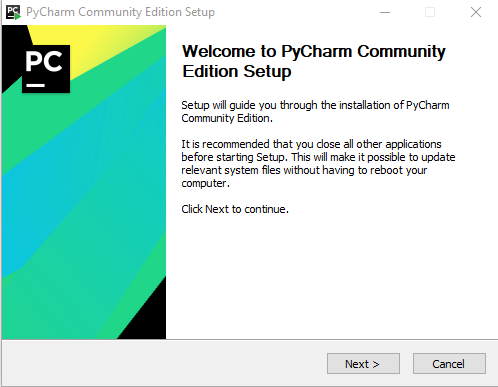
- Hỗ trợ cho các khung công tác web: Django, web2py và Flask.

- Phát triển Python của Google App Engine.

- Tích hợp kiểm soát phiên bản: giao diện người dùng thống nhất cho Mercurial, Git, Subversion, Perforce và CVS với các thay đổi và hợp nhất.

#### 1.6.2. Các bước cài đặt PyCharm

Sau khi tải file cài đặt, bấm chọn vào file đã tải, hộp thoại sau xuất hiện.

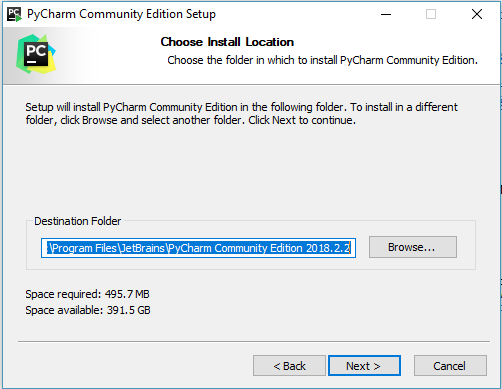
Bấm chọn Next như hình bên dưới.**

**Hình 1.12.Cài đặt PyCHarm**

Hộp thoại sau xuất hiện.

Chọn nơi lưu phần mềm.

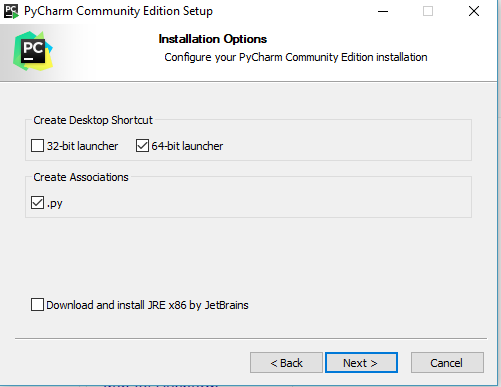
Sau đó, chọn Next.



**Hình 1.13.Cài đặt PyCHarm**

Sau đó chọn mục 64-bit launcher (đối với máy 64 bit), chọn mục 32-bit launcher (đối với máy 32 bit).

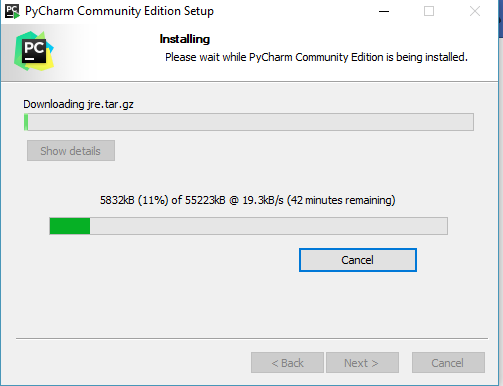
Sau đó bấm Next.



**Hình 1.14.Bấm Next**

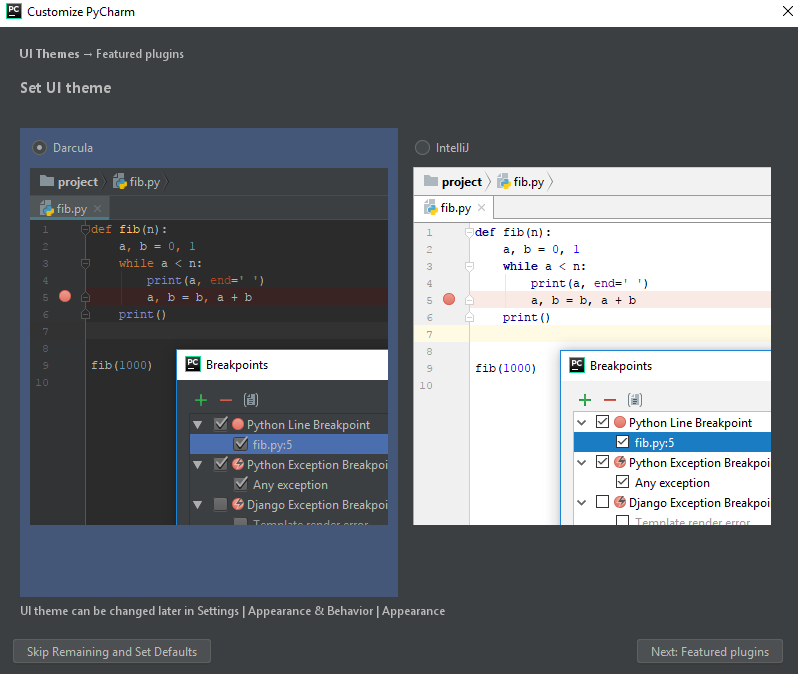
Chờ đợi cho quá trình tải hoàn tất.

Sau khi tải xong, bấm Next.



**Hình 1.15.Tiếp tục bấm next để cài đặt**

Chọn giao diện sử dụng:

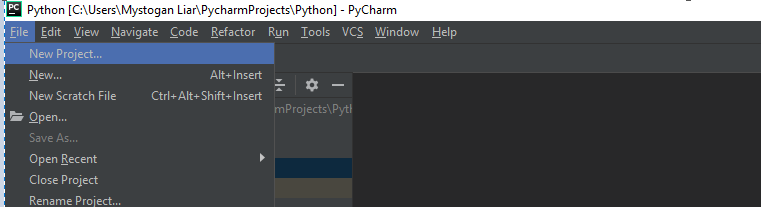


**Hình 1.16.Chọn giao diện sử dụng PyCharm**

#### 1.6.3. Hướng dẫn tạo project Python trên phần mềm Pycharm

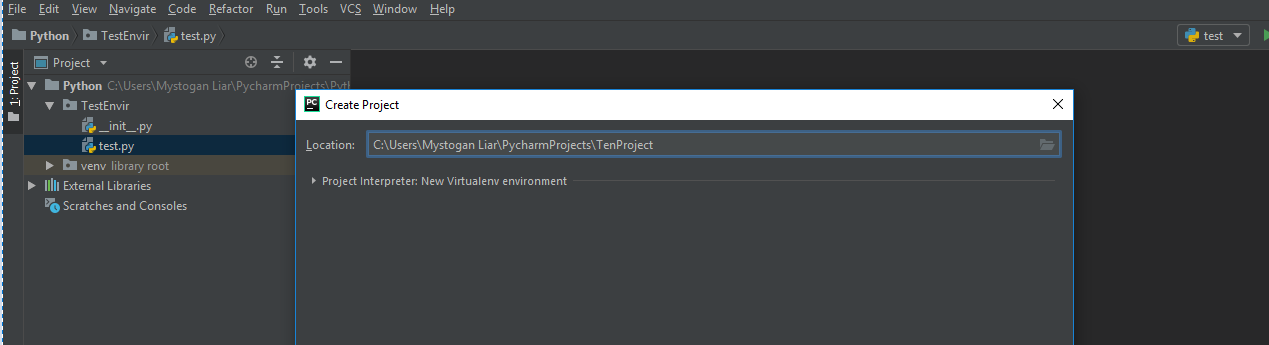
**Bước 1:** Tạo 1 project python trên Pycharm.

Chọn File -> New Project.



**Hình 1.17.Tạo project trên PyCharm**

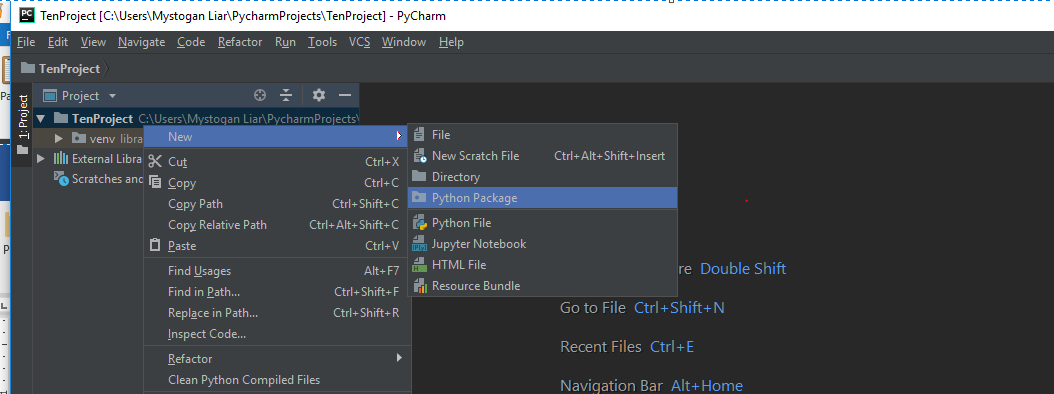
**Bước 2:** Đặt tên Project.

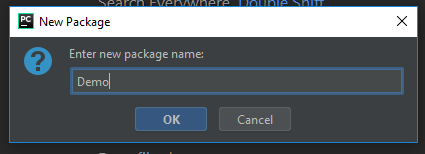


**Hình 1.18.Đặt tên project**

Sau đó chọn “OK ” để khởi tạo Project.

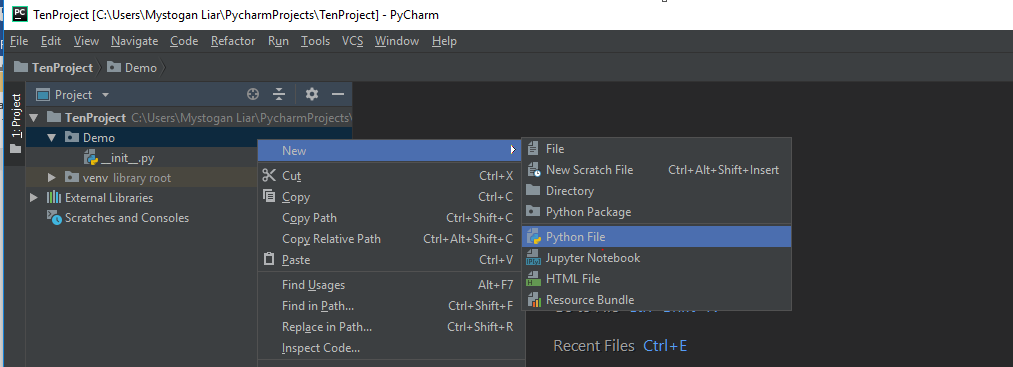
**Bước 3:** Tiến hành tạo package cho project như hình và đặt tên cho nó.



**Hình 1.19.Tạo package**

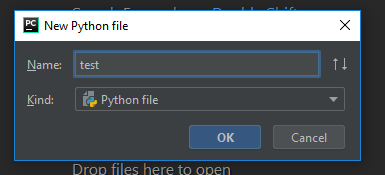
**Hình 1.20.đặt tên package**

**Bước 4:** Sau khi tạo package t tiến hành tạo file python để code (click phải vào package ‘Demo’).

New -> Python File.

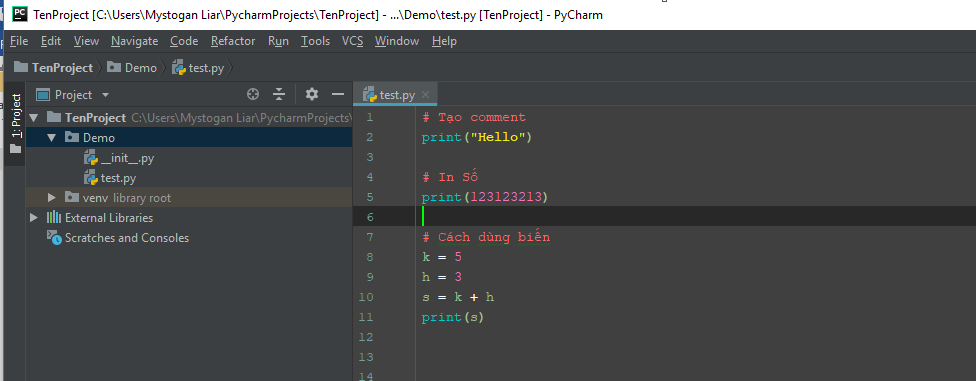
**Hình 1.21.Tạo File code**

Sau đó chỉ cần đặt tên và kết thúc quá trình tạo 1 file python.



**Hình 1.22.Đặt tên File**

**Bước 5:** Tiến hành lập trình.



**Hình 1.23.Tiến hành lập trình**

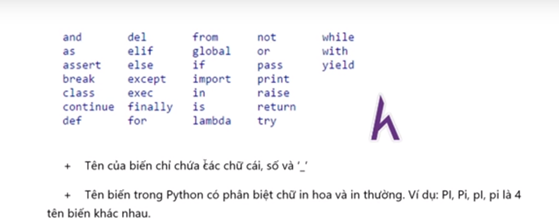
Lưu ý

- Python hỗ trợ cả kiểu dữ liệu số phức và phân số.

- Dùng dấu # làm phương thức dùng để comment.

- Nếu là biến chuỗi phải đặt trong dáu nháy kép: VD: ten= “CPAR”.

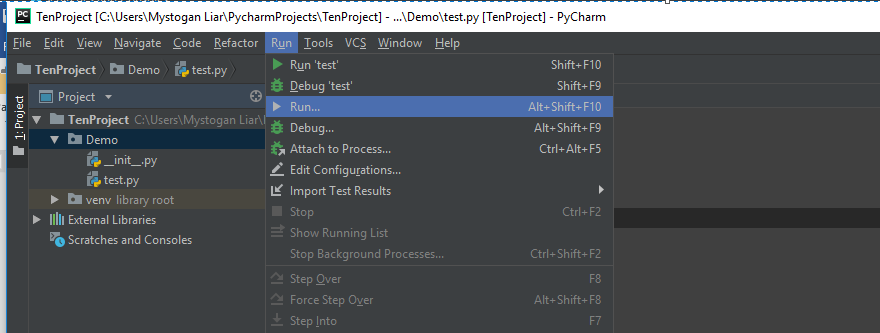
- Các từ khóa trong python cần tránh khi đặt tên biến.



**Hình 1.24.Các tên biến cần tránh trong python**

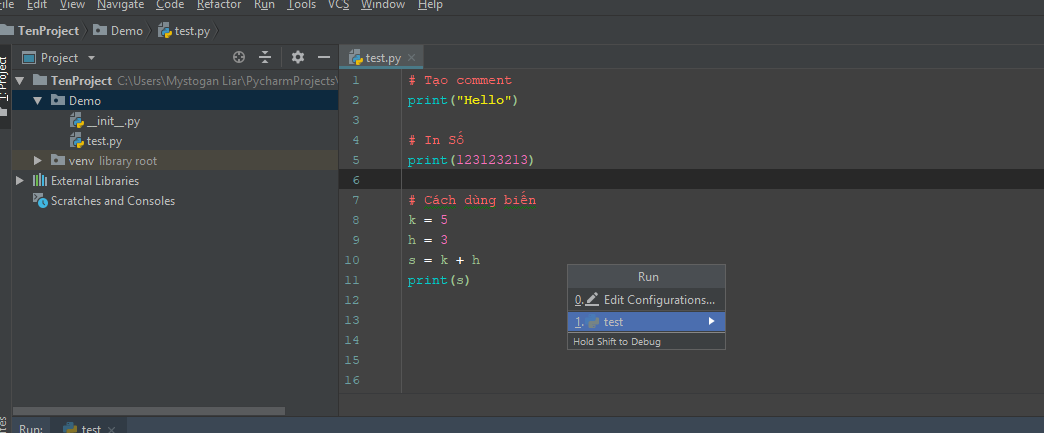
**Bước 6:** Cách chạy 1 file code.

Ấn ‘Run’ và ‘Run’ như trong hình sau:



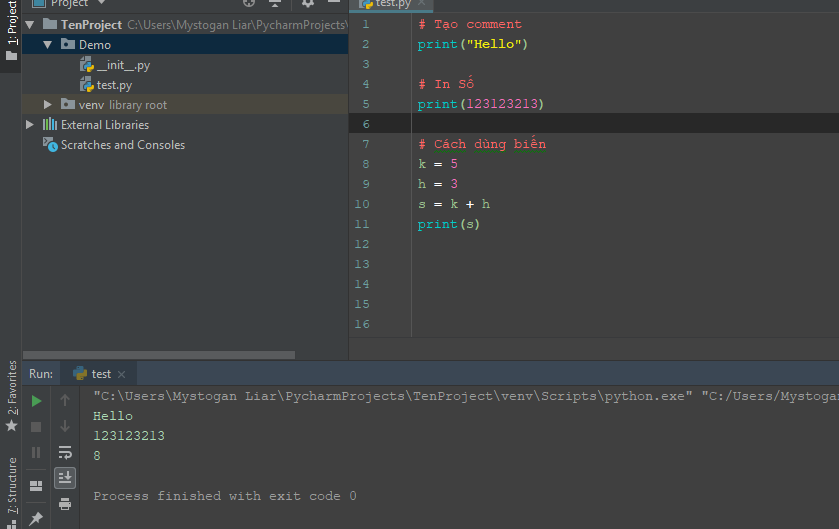
**Hình 1.25.Chạy file code**

Sau đó chọn File cần chạy.



**Hình 1.26.Chọn File**

Kết quả sau khi chạy file:



**Hình 1.27..Hiển thị kết quả**

Chương trình sẽ trả về kết quả và số lỗi của file code (ở hình trên là 0 lỗi).

## **Chương 2: KHAI PHÁ LUẬT KẾT HỢP**

### 2.1. BÀI TOÁN KHAI PHÁ LUẬT KẾT HỢP

Bài toán giỏ mua hàng trong siêu thị.

Giả sử chúng ta có rất nhiều mặt hàng, ví dụ như “bia”, “sữa”, … (coi là tính chất hoặc trường). Khách hàng khi đi siêu thị sẽ bỏ vào giỏ mua hàng của mình một số mặt hàng nào đó, và chúng ta muốn tìm hiểu các khách hàng thường mua các mặt hàng nào đồng thời, chúng ta không cần biết khách hàng cụ thể là ai. Nhà quản lý dùng những thông tin này để điều chỉnh việc nhập hàng về siêu thị, hay đơn giản là để bố trí sắp xếp các mặt hàng gần nhau, hoặc bán các mặt hàng đó theo một gói hàng, giúp cho khắc đỡ mất công tìm kiếm.

Khai phá luật kết hợp được mô tả như sự tương quan của các sự kiện những sự kiện xuất hiện thường xuyên một các đồng thời. Nhiệm vụ chính của khai phá luật kết hợp là phát hiện ra các tập con cùng xuất hiện trong một khối lượng giao dịch lớn của một cơ sở dữ liệu cho trước.

### 2.2. MỘT SỐ KHÁI NIỆM CƠ BẢN

*Item (phần tử)*

- Các phần tử, mẫu, đối tượng đang được quan tâm.

- J = {I1, I2, …, Im}: tập tất cả m phần tử có thể có trong tập dữ liệu.

*Itemset (tập phần tử)*

- Tập hợp các items.

- Một itemset có k items gọi là k-itemset.

*Transaction (giao dịch)*

- Lần thực hiện tương tác với hệ thống (ví dụ: giao dịch “khách hàng mua hàng”).

- Liên hệ với một tập T gồm các phần tử được giao dịch.

*Association (sự kết hợp) và association rule (luật kết hợp)*

- Sự kết hợp: các phần tử cùng xuất hiện với nhau trong một hay nhiều giao dịch.

- Thể hiện mối liên hệ giữa các phần tử/các tập phần tử.

- Luật kết hợp: qui tắc kết hợp có điều kiện giữa các tập phần tử.

- Thể hiện mối liên hệ (có điều kiện) giữa các tập phần tử.

- Cho A và B là các tập phần tử, luật kết hợp giữa A và B là A -> B.

- B xuất hiện trong điều kiện A xuất hiện.

*Support (độ hỗ trợ)*

- Độ đo đo tần số xuất hiện của các phần tử/tập phần tử.

- Minimum support threshold (ngưỡng hỗ trợ tối thiểu).

- Giá trị support nhỏ nhất được chỉ định bởi người dùng.

*Confidence (độ tin cậy)*

- Độ đo đo tần số xuất hiện của một tập phần tử trong điều kiện xuất hiện của một tập phần tử khác.

- Minimum confidence threshold (ngưỡng tin cậy tối thiểu).

- Giá trị confidence nhỏ nhất được chỉ định bởi người dùng.

*Frequent itemset (tập phần tử phổ biến)*

- Tập phần tử có support thỏa minimum support threshold.

- Cho A là một itemset.

- A là frequent itemset nếu support(A) >= minimum support threshold.

#### 2.2.1. Literal

Một literal p là một cặp thuộc tính có dạng (Ai,v), trong đó Ai là một thuộc tính còn v là một giá trị. Một tuple t thỏa mãn literal p = (Ai, v) nếu và chỉ nếu ti=v, trong đó ti là giá trị của thuộc tính thứ i của tuple t.

#### 2.2.2. Định nghĩa luật kết hợp

Một luật r có dạng “p1 ^ p2 ^…^ pi  🡪 c”, bao gồm tổ hợp các literal p1, p2,…,pi được liên kết lại với nhãn lớp c. Một tuple t thỏa mãn quy tắc vế trái của luật r nếu và chỉ nếu nó thỏa mãn mọi literal trong luật, r sẽ dự đoán rằng t thuộc lớp c. Nếu một quy tắc chứa literal rỗng, thì nó thỏa mãn bất cứ bộ dữ liệu nào.

### 2.2. CÁC THUẬT TOÁN CƠ BẢN

#### 2.2.1. Thuật toán Apriori

Apriori là thuật toán được đề xuất bởi *R. Agrawal và R. Srikant* vào năm 1993 để khai thác các tập item đối với các luật kết hợp kiểu bool.

Apriori dùng cách tiếp cận lặp được biết đến như tìm kiếm *level-wise*, với các tập *k* item được dùng để thăm dò các tập (*k*+1) item. Đầu tiên, tập các tập 1 item phổ biến được tìm thấy bằng cách quét cơ sở dữ liệu để đếm số lượng từng item, và thu thập những item thỏa mãn độ hỗ trợ tối thiểu[.[1]](#tài_liệu_1)

Hạn chế của thuật toán Apriori.

- Số lượng lớn tập phổ biến được tạo ra làm gia tăng sự phức tạp không gian.

- Quá nhiều lần duyệt cơ sở dữ liệu được yêu cầu vì số lượng lớn tập phổ biến được tạo.

- Khi số lần duyệt cơ sở dữ liệu nhiều làm gia tăng sự phức tạp thời gian khi cơ sở dữ liệu gia tăng.

#### 2.2.2. Thuật toán Apriori-TID

Thuật toán Apriori-TID là phần mở rộng theo hướng tiếp cận cơ bản của thuật toán Apriori. Thay vì quét toàn bộ cơ sở dữ liệu để tìm độ hỗ trợ (*support*) của các tập ứng với mỗi bước gây lãng phí tài nguyên, khó triển khai đối với các cơ sỡ dữ liệu lớn thì thuật toán Apriori-TID chỉ duyệt toàn bộ cơ sỡ dữ liệu lần đầu tiên để tìm độ hỗ trợ cho một phần tử (item), từ bước thứ hai trở đi thuật toán Apriori-TID tạo ra thêm một bộ lưu trữ song song cả giao dịch và tập ứng viên, bộ này sẽ dùng để đếm độ hỗ trợ của mỗi ứng viên thay cho cơ sở dữ liệu như ở Apriori. Nhờ đó tránh phải duyệt toàn bộ cơ sở dữ liệu nhiều lần.[[1]](#tài_liệu_1)

#### 2.2.3. Logistics Regression

Logistics Regression là nghiên cứu mối tương quan giữa một (hay nhiều) yếu tố, nguy cơ (*risk factor*) và đối tượng phân tích (*outcome*).[[2]](#tài_liệu_2)

Logistics Regression được phát triển bởi nhà thống kê [*David Cox*](https://en.wikipedia.org/wiki/David_Cox_(statistician)) vào năm 1958. Mô hình logistic nhị phân được sử dụng để ước lượng xác suất của một phản hồi nhị phân dựa trên một hoặc nhiều biến (hoặc tính độc lập) dự báo. Nó cho phép một người nói rằng sự hiện diện của một yếu tố nguy cơ làm tăng tỷ lệ cược của một kết quả nhất định bởi một yếu tố cụ thể.

#### 2.2.4. Support Vector Machine (SVM) cho bài toán phân lớp quan điểm

Phương pháp Support Vector Machine (SVM) được coi là công cụ mạnh cho những bài toán phi tuyến tính được các tác giả *Vapnik và Chervonenkis* phát triển mạnh mẽ năm 1995. Phương pháp này dựa trên nguyên lý Cực tiểu hóa rủi ro có cấu trúc SRM (*Structural Risk Minimization*), được xem là một trong các phương pháp phân lớp giám sát không tham số tinh vi nhất cho đến hiện nay. Các hàm công cụ của SVM cho phép tạo không gian chuyển đổi để xây dựng mặt phẳng phân lớp[.[3]](#tài_liệu_3)

SVM là phương pháp dựa trên nền tảng của lý thuyết thống kê nên có một nền toán học chặt chẽ để đảm bảo rằng kết quả tìm được là chính xác.

SVM là thuật toán giám sát được sử dụng cho phân lớp dữ liệu.

Phân lớp quan điểm là xử lý một tập hợp các kết quả cho một đối tượng nhất định sinh ra một danh sách các thuộc tính và các quan điểm về chúng.

## **Chương 3: PHÂN LỚP DỰA TRÊN LUẬT KẾT HỢP**

### 3.1. TỔNG QUAN VỀ PHÂN LỚP DỰA TRÊN LUẬT KẾT HỢP

Nhiều ứng dụng quan trọng như chẩn đoán y tế, phân tích văn bản, lừa đảo trang web và nhiều ứng dụng khác cần một công cụ tự động nhân tạo để tăng cường quá trình ra quyết định. Sử dụng các quy tắc kết hợp trong quy trình phân loại là một kỹ thuật trong lĩnh vực khai thác dữ liệu để đưa ra các quyết định chính xác và quan trọng. Và kỹ thuật này được gọi là association classification (AC). Thông thường qua hai giai đoạn

Giai đoạn 1: Tìm luật kết hợp

Giai đoạn 2: Phân lớp dựa trên luật kết hợp đã tìm được

### 3.2. CÁC CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN

#### 3.2.1. FOIL (first order inductive leaner)

FOIL được đề xuất và phát triển bởi *Quinlan* (*Quinlan*, 1993). FOIL là một thuật toán tham lam mà các luận học của nó dùng để phân biệt các mẫu dương (tích cực) từ những mẫu âm (tiêu cực). FOIL liên tục nghiên cứu luật tốt nhất hiện tại và loại bỏ tất cả các mẫu dương (tích cực) được đảm bảo bởi luật cho đến khi tất cả những mẫu dương (tích cực) trong bộ dữ liệu được đảm bảo (bảo vệ). Đầu vào của Foil gồm các tiền điều kiện và các kết luận. Đầu ra là một tập các luật sinh từ các tiền điều kiện và các kết luận đó. Thuật toán FOIL được trình bài như sau

Đối với các vấn đề đa lớp, FOIL được áp dụng trên từng lớp. Đối với mỗi lớp, những mẫu của nó được dùng như là mẫu dương (*tích cực*)1 và những lớp khác như là những mẫu âm *(tiêu cực*)2. Những quy luật đối với tất cả các lớp thì được hợp nhất với nhau để tạo thành một bộ kết quả.

Khi lựa chọn các ký tự, Foil Gain được sử dụng để đo lường thông tin thu được từ việc thêm ký tự đó vào luật hiện tại.

Ví dụ về mẫu âm, mẫu dương: Như trong chuẩn đoán bệnh nhân bị HIV, tập hợp những người bị chuẩn đoán mắc bệnh được dán nhãn 1, đây là mẫu dương; những người được chuẩn đoán không bị bệnh sẽ được dán nhãn 0, tập những người như thế sẽ là tập mẫu âm.

Trường hợp này ta quan tâm đến những người bị chuẩn đoán bị mắc bệnh vì số người bị chuẩn đoán bị mắc bệnh sẽ ít hơn người không mắc bệnh, ta sẽ quan tâm đến độ tuổi, khu vực sinh sống, ngành nghề để tìm ra các luật kết hợp tốt hơn nên sẽ dán nhãn 1 cho những người này và làm các phép toán trên đó thay vì tập mẫu gồm tất cả những người đến xét nghiệm.

* **Ưu điểm:**

Thuật toán nhanh, đơn giản, hiệu quả trong không gian nhỏ.

* **Nhược điểm:**

Foil không đạt được kết quả cao, chính xác là nó tạo ra số lượng rất nhỏ các qui tắc và có nhiều luật có độ chính xác thấp.

Foil có một hạn chế là luật qui nạp không được chứa bất kỳ ký hiệu hằng hoặc ký hiệu biến nào (ví dụ: ta không được viết *color (X, red)*, mà phải viết là *color(X), red(Y)*).

Foil sử dụng thuật toán leo đồi *(Hill Climbing)* - kỹ thuật tối ưu toán học thuộc họ tìm kiếm cục bộ. Nó thực hiện tìm một trạng thái tốt hơn trạng thái hiện tại để mở rộng, nên khó tìm thấy trạng thái đích trong không gian big data.[[4]](#tài_liệu_7)

**3.2.2. Thuật toán phân lớp kết hợp sử dụng tối ưu hóa đàn kiến (ACO)**

Dựa vào tính chất của loài kiến khi đi tìm đường kiếm mồi, thuật toán ACO được đề xuất bởi Marco Dorigo vào năm 1992. Thuật toán này được phát triển bằng cách mô hình hóa một số khía cạnh của hành vi tìm kiếm thức ăn của kiến. Kiến truyền thông tin trên con đường mà chúng theo dõi nằng cách đặt một chất hóa học trên con đường mà chúng theo dĩ bằng cách đặt 1 chất hóa học gọi là pheromone vào môi trường. Những con kiến khác sẽ có khả năng đi trên con đường có nồng độ pheromone cao hơn sau đó là những con đường có nồng độ pheromone thấp hơn. Hiện tượng này được mô hình hóa trong thuật toán ACO.

Mỗi một con kiến sẽ tạo ra một giải pháp thu thập các giải pháp để tạo ra 1 giải pháp hoàn chỉnh. Khi một giải pháp được xây dựng, chất lượng kiếm mồi cũng được xác định, và các nồng độ pheromone tỷ lệ thuận với chất lượng kiếm mồi đó. Những giải pháp đường đi có nồng độ pheromone cao hơn được xác định là góp phần giải pháp tốt và liên tục xuất hiện trong các giải pháp.

ACO đã được áp dụng thành công để giải quyết nhiều vấn đề về tối ưu hóa như lập công việc, các vấn đề về tập hợp, phân công và định tuyến mạng.

* Ưu điểm: Thuật toán đơn giản dễ cài đặt.
* Khuyết điểm: Thuật toán chỉ phù hợp với tập dữ liệu lớn, đối với tập có dữ liệu nhỏ thì độ chính xác không cao.

Phương pháp ACO được coi là phương pháp tốt nhất trong việc phân loại với độ chính xác cao. [[5]](#tài_liệu_8)

#### 3.2.3. Cải tiến thuật toán CAR (An Efficient Algorithm for Mining ClassAssociation Rules with Interestingness Measures)

Một cải tiến mới của thuật toán CAR đó chính là áp dụng cấu trúc cây, việc áp dụng này làm giảm đi đáng kể thời gian thực hiện do tính linh loạt của cấu trúc cây.

Thuật toán sử dụng cấu trúc cây cho việc chứa thông tin vào các nút, điều này có ích cho việc tăng tốc tạo ra luật. Với obidset chứa các đối tượng cần thiết trong itemset, chúng có thể lấy độ support của itemset mới một cách nhanh chóng bằng cách tính giao điểm của hai obidset, obidset cho ta tính toán số lượng của từng lớp. Với các số đếm này, thuật toán có thể dễ dàng xác định các quy tắc có giá trị đo cao nhất. [[6]](#tài_liệu_10)

#### 3.2.4. Phân lớp nhanh dựa trên luật kết hợp (Fast Classification Based on Association Rules Algorithm)

Thuật toán này được sinh ra nhằm mục đích như chẩn đoán y tế, phân tích văn bản, lừa đảo trang web…, cần một công cụ để tăng cường quá trình qua quyết định.

Theo thuật toán truyền thống là AC (association classsification) nó có nhược điểm bị ảnh hưởng bởi kích thước dữ liệu, 1 số thuật toán AC có độ chính xác cao nhưng về mặt thời gian thì mất rất nhiều để cây dựng mô hình. Để giải quyết vấn đề này, Thuật toán phân lớp nhanh dựa trên luật kết hợp (FCBA) ra đời, dựa trên các phương pháp cắt tỉa bên trong và bên ngoài để tạo ra luật bằng thuật toán apriori nâng cao.

Bằng cách so sánh thuật toán FCBA [[7]](#tài_liệu_12) so với các thuật toán CBA, CMAR, MCAR và FACA dựa trên 11 bộ dữ liệu UCI. Hầu hết là dữ liệu y tế có kích thước khác nhau đã đánh giá được khả năng mở rông, độ chính xác và thời gian xây dựng của FCBA là tốt nhất so với các thuật toán vừa nêu.

FCBA nhằm mục đích tối ưu hóa thời gian xây dựng mô hình và tỷ lệ chính xác phân loại cho thuật toán CBA tạo ra các quy tắc sử dụng apriori, FCBA sử dụng một bẳng xếp mới phương pháp nâng cao tốc độ của thuật toán apriori bằng cách thêm 1 cơ chế cắt tỉa nội bộ hiệu quả mới cho quy trình tạo luật. Dựa vào độ xuất hiện thường xuyên của luật được tạo ra, xếp hạng sẽ được đề xuất cắt tỉa.

- Ưu điểm: Thuật toán FCBA vượt trội so với các thuật toán khác về độ chính xác và thời gian để xây dựng mô hình. FCBA được xếp hạng đầu tiên trong 64% và thứ hai trong 36% bộ dữ liệu, với thời gian trung bình dưới 0,01 giây. Do đó, nó đạt được độ chính xác cao nhất và thời gian trung bình nhanh nhất để xây dựng mô hình, so với các thuật toán khác.

- Nhược điểm: do cắt tỉa trong quá trình tìm luật nên gây khó khăn trong việc cài đặt.

#### 3.2.5. Khai thác quy tắc kết hợp (ARM)

Khai thác quy tắc kết hợp (ARM) [[8]](#tài_liệu_12) là một chủ đề nghiên cứu quan trọng nhất và nó được sử dụng phổ biến trong các lĩnh vực y tế và các lĩnh vực khác. Kể từ khi được giới thiệu vào năm 1993, quy tắc kết hợp đã nhận được rất nhiều sự chú ý. Nó được định nghĩa là mối quan hệ giữa các mục. Quá trình tìm kiếm các quy tắc hiệp hội đã đạt được sự nổi bật. Nó vẫn là một trong những kỹ thuật khám phá nổi bật trong khám phá tri thức và khai thác dữ liệu. Mục tiêu của ARM là khám phá các mẫu, liên kết thường xuyên giữa các bộ vật phẩm trong cơ sở dữ liệu giao dịch hoặc các kho dữ liệu khác. Các quy tắc của hiệp hội được sử dụng rộng rãi trong các lĩnh vực khác nhau như mạng viễn thông, quản lý rủi ro thị trường và thị trường, kiểm soát hàng tồn kho, …

Hai kỹ thuật khai thác dữ liệu cấp cao là mô tả *(Descriptive)* và dự đoán *(Predictive).*

***Ưu điểm***

Ít tốn thời gian vì không yêu cầu quét cơ sở dữ liệu mỗi lần.

***Nhược điểm***

Thời gian chạy thuật toán và bộ nhớ khác nhau cho các bộ dữ liệu khác nhau.

### 3.3. THUẬT TOÁN PRM (PREDICTIVE RULE MINING)

#### 3.3.1. Tổng quan

Phương pháp PRM là cả tiến của FOIL và được đề xuất và phát triển bởi *Quinlan* *(Quinlan and Cameron-Jones 1993).* [[1]](#tài_liệu_1)

Đối với các vấn đề đa lớp, FOIL được áp dụng trên từng lớp. Đối với mỗi lớp, những mẫu của nó được dùng như là mẫu dương (*tích cực*)[1] và những lớp khác như là những mẫu âm (*tiêu cực*). Những quy luật đối với tất cả các lớp thì được hợp nhất với nhau để tạo thành một bộ kết quả.

Khi lựa chọn các ký tự, FOIL Gain được sử dụng để đo lường thông tin thu được từ việc thêm ký tự đó vào luật hiện tại.

Phương pháp PRM (Xiaoxin Yin and Jiawei Han 2003) được sinh ra dựa trên FOIL, một thuật toán trong đó đổi trọng số để đạt được độ chính xác cao hơn và hiệu quả. Một lý do đó FOIL không đạt được độ chính xác cao là nó tạo ra một số có độ lợi rất nhỏ của luật. Trong PRM, sau khi một item được tìm ra bởi một luật, thay vì loại bỏ nó, trọng lượng của nó là giảm.

#### 3.3.2. Giới thiệu tập Positive and Negative.

Positve và Negative là tập tích cực và tiêu cực tương ứng với việc phân lớp.

Ví dụ: Trong chuẩn đoán bệnh nhân bị HIV, tập hợp những người bị chuẩn đoán mắc bệnh được đưa vào tập P, những người được chuẩn đoán không bị bệnh sẽ được đưa vào tập N. Ta quan tâm đến những người bị chuẩn đoán bị mắc bệnh trong tập P theo độ tuổi, khu vực sinh sống, ngành nghề để tìm ra luật trả lời cho giả thiết việc bị nhiễm HIV.

#### 3.3.3. Thuật toán

**Input**: Tập hợp dữ liệu được chia thành 2 mẫu P và N.

**Output**: Danh sách các quy tắc dựa trên phân lớp của thuộc tính.

* Thuật toán

- Chú thích:

TotalWeight : tổng trọng số

Set rule : tập nguyên tắc

* Vấn đề:

Trọng số mỗi item P và N đều bằng 1 để xét độ quan trong ban đầu đều bằng nhau.

* Tổng quan luồng thuật toán [3]

Set rule R = rỗng

Total Weight =TotalWeight(P)

While [TotalWeight(P) > λ\*totalWeight] // λ= 0.05 (vòng lặp 1)

N’ = N, P’ = P, PN’ = PN

Rule r = Set rule rỗng

While true (vòng lặp 2)

Tìm g = gain(p) có giá trị lớn nhất

if g < = MIN\_GAIN // MIN\_GAIN = 0.7

break

Chèn p vô r

For [mỗi item t trong P’, N’ không thuộc r]

Xoá t từ P’ hoặc N’

if N’ = null

break

end

R = R Union {r}

for [mỗi item t trong P thuộc r]

t.weight = α(t.weight) // α =1/3

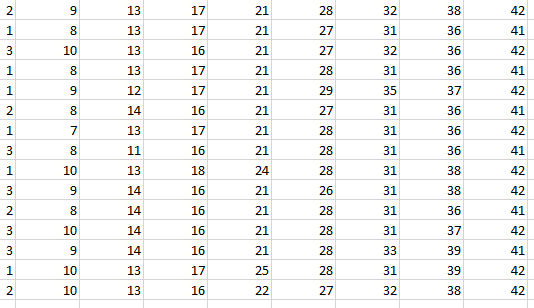
end

Return R

#### 3.3.4. Ví dụ

Lấy 15 dòng trong tập Pima Indian set (Cơ sở dữ liệu bệnh tiểu đường Pima Indians).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |



**Hình 3.1. Ví dụ mẫu thuộc tính**

**Trong đó**

# Cột 1. Số lần mang thai (đơn vị là lần).

# Cột 2. Nồng độ glucose huyết tương 2 giờ trong một thử nghiệm dung nạp glucose đường uống.

# Cột 3. Huyết áp tâm trương (mm/Hg).

# Cột 4. Triceps da gấp độ dày (mm).

# Cột 5. 2-giờ huyết thanh insulin (mu U / ml).

# Cột 6. Chỉ số khối cơ thể (trọng lượng tính bằng kg / (chiều cao tính theo m) ^ 2).

# Cột 7. Chức năng phả hệ tiểu đường.

# Cột 8. Tuổi (năm).

# Cột 9. Biến lớp (có mắc bệnh hoặc không mắc bệnh là 41, không mắc là 42).

BƯỚC 1:

Phân lớp 41 và 42: ta sẽ bắt đầu với trọng số = 1.

+ Tập P: Số người không bị mắc bệnh tiểu đường

**Bảng 3.2. Tập Positive bệnh tiểu đường***.*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Số lần mang thai | Nồng độ glucose | Huyết áp (mm HG) | Độ dày da (mm) | Huyết thanh insulin(mu U/ml) | Chỉ số khối cơ thể (kg/m^2) | Chức năng phả hệ tiểu đường | Tuổi (năm) | Biến phân lớp |  |
| 1 | 8 | 13 | 17 | 21 | 27 | 31 | 36 | 41 | 1 |
| 1 | 8 | 13 | 17 | 21 | 28 | 31 | 36 | 41 | 1 |
| 2 | 8 | 14 | 16 | 21 | 27 | 31 | 36 | 41 | 1 |
| 3 | 8 | 11 | 16 | 21 | 28 | 31 | 36 | 41 | 1 |
| 2 | 8 | 14 | 16 | 21 | 28 | 31 | 36 | 41 | 1 |
| 3 | 9 | 14 | 16 | 21 | 28 | 33 | 39 | 41 | 1 |

+ Tập N: Số người bị mắc bệnh tiểu đường

**Bảng 3.3. Tập Negative bệnh tiểu đường**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Số lần mang thai | Nồng độ glucose | Huyết áp (mm HG) | Độ dày da (mm) | Huyết thanh insulin(mu U/ml) | Chỉ số khối cơ thể (kg/m^2) | Chức năng phả hệ tiểu đường | Tuổi (năm) | Biến phân lớp |  |
| 2 | 9 | 13 | 17 | 21 | 28 | 32 | 38 | 42 | 1 |
| 3 | 10 | 13 | 16 | 21 | 27 | 32 | 36 | 42 | 1 |
| 1 | 9 | 12 | 17 | 21 | 29 | 35 | 37 | 42 | 1 |
| 1 | 7 | 13 | 17 | 21 | 28 | 31 | 36 | 42 | 1 |
| 1 | 10 | 13 | 18 | 24 | 28 | 31 | 38 | 42 | 1 |
| 3 | 9 | 14 | 16 | 21 | 26 | 31 | 38 | 42 | 1 |
| 3 | 10 | 14 | 16 | 21 | 28 | 31 | 37 | 42 | 1 |
| 1 | 10 | 13 | 17 | 25 | 28 | 31 | 39 | 42 | 1 |
| 2 | 10 | 13 | 16 | 22 | 27 | 32 | 38 | 42 | 1 |

Xác định PN array

Tính trọng số của từng thuộc tính trong A lần lượt với P và N.

Vd: value 1 trong cột thuộc tính Số lần mang thai có trọng số p là 2, n là 4

Ta có mảng PN:

**Bảng 3-4. Mảng PN sau khi tính số lần xuất hiện**

|  |  |  |
| --- | --- | --- |
| Value (a) | Trọng số của a trong P | Trọng số của a trong N |
| 1 | 2 | 4 |
| 2 | 2 | 2 |
| 3 | 2 | 3 |
| 4 | 0 | 0 |
| 5 | 0 | 0 |
| 6 | 0 | 0 |
| 7 | 0 | 1 |
| 8 | 5 | 0 |
| 9 | 1 | 3 |
| 10 | 0 | 5 |
| 11 | 1 | 0 |
| 12 | 0 | 1 |
| 13 | 2 | 6 |
| 14 | 3 | 2 |
| 15 | 0 | 0 |
| 16 | 4 | 4 |
| 17 | 2 | 4 |
| 18 | 0 | 1 |
| 19 | 0 | 0 |
| 20 | 0 | 0 |
| 21 | 6 | 6 |
| 22 | 0 | 1 |
| 23 | 0 | 0 |
| 24 | 0 | 1 |
| 25 | 0 | 1 |
| 26 | 0 | 0 |
| 27 | 2 | 2 |
| 28 | 4 | 5 |
| 29 | 0 | 1 |
| 30 | 0 | 0 |
| 31 | 5 | 5 |
| 32 | 0 | 3 |
| 33 | 1 | 0 |
| 34 | 0 | 0 |
| 35 | 0 | 1 |
| 36 | 5 | 2 |
| 37 | 0 | 2 |
| 38 | 0 | 4 |
| 39 | 1 | 1 |
| 40 | 0 | 0 |

**Bước 2**:

- Tính ngưỡng tổng trọng số (TWT) là:

TWT = totalWeight\*TOTAL\_WEIGHT\_FACTOR

= 6.0\*0.05 = 0.3

Trong đó:

TotalWeight: tổng trọng số trong P = 6

Total-Weight-Factor = λ = 0.05

* Xét tập nguyên tắc R = Φ

**Bước 3**: Bước vô vòng lặp: While [TotalWeight(P) > λ\*totalWeight (vòng lặp 1)

Trong khi TotalWeight(P) > TWT

* Copy P, N, PN tương ứng với P’, N’, PN’.
* Tập luật r = Φ

**Bước 4**: Bước vô vòng lặp: While true (vòng lặp 2)

* Tính gain(a) tương ứng P theo A và lưu kết quả vào A

Trọng số của N là 9 (chú ý rằng sau này nó sẽ là hằng số cho lớp trong câu hỏi). Sau đó tiến hành thiết lập lại P cho đến khi tổng trọng số của nó nhỏ hơn ngưỡng tổng trọng số (TWT).

Trong đó: WP’: *tổng số lần xuất hiện của a trong P’*

WN’: *tổng số lần xuất hiện của a trong N’*

WP: *tổng trọng số của P*

WN: *tổng trọng số của N*

*PN = WP + WN*

Ta có bảng giá trị Gain của trong P như sau:

**Bảng 3-5. Tính độ lợi**

|  |  |
| --- | --- |
| Value (a) | Gain(a) |
| 1 | -0.11598 |
| 2 | 0.13389 |
| 3 | 0.29226 |
| 4 | 0 |
| 5 | 0 |
| 6 | 0 |
| 7 | 0 |
| 8 | 1.21519 |
| 9 | -0.1549 |
| 10 | 0 |
| 11 | 0.243038 |
| 12 | 0 |
| 13 | -0.3098 |
| 14 | 0.354298 |
| 15 | 0 |
| 16 | 0.267787 |
| 17 | 0 |
| 18 | 0 |
| 19 | 0 |
| 20 | 0 |
| 21 | 0.401681 |
| 22 | 0 |
| 23 | 0 |
| 24 | 0 |
| 25 | 0 |
| 26 | 0 |
| 27 | 0.133894 |
| 28 | 0.128739 |
| 29 | 0 |
| 30 | 0 |
| 31 | 0.334734 |
| 32 | 0 |
| 33 | 0.243038 |
| 34 | 0 |
| 35 | 0 |
| 36 | 0.819284 |
| 37 | 0 |
| 38 | 0 |
| 39 | 0.066947 |
| 40 | 0 |

* Tìm gain(a) tốt nhất theo A

Thuộc tính với độ lợi lớn nhất là 8 (gain (8) = 1.21519 > MIN\_GAIN) => Thêm thuộc tính này vào trong tập r: {8}-> {41} và làm tiếp bước 5.

**Bước 5**: Tất cả các item t nào không chứa value 8 sẽ được xoá khỏi P’, N’.

Thay đổi mảng PN’ theo việc loại bỏ t.

Ta có:

P’:

{1 8 13 17 21 27 31 36 41}

{1 8 13 17 21 28 31 36 41}

{2 8 14 16 21 27 31 36 41}

{3 8 11 16 21 28 31 36 41}

{2 8 14 16 21 28 31 36 41}

N’: Null

Sau đó ta có mảng PN’:

**Bảng 3-6. Xác định lại mảng PN**

|  |  |  |
| --- | --- | --- |
| Value (a) | Trọng số của a trong P | Trọng số của a trong N |
| 1 | 2 | 0 |
| 2 | 2 | 0 |
| 3 | 1 | 0 |
| 4 | 0 | 0 |
| 5 | 0 | 0 |
| 6 | 0 | 0 |
| 7 | 0 | 0 |
| 8 | 5 | 0 |
| 9 | 0 | 0 |
| 10 | 0 | 0 |
| 11 | 1 | 0 |
| 12 | 0 | 0 |
| 13 | 2 | 0 |
| 14 | 2 | 0 |
| 15 | 0 | 0 |
| 16 | 3 | 0 |
| 17 | 2 | 0 |
| 18 | 0 | 0 |
| 19 | 0 | 0 |
| 20 | 0 | 0 |
| 21 | 5 | 0 |
| 22 | 0 | 0 |
| 23 | 0 | 0 |
| 24 | 0 | 0 |
| 25 | 0 | 0 |
| 26 | 0 | 0 |
| 27 | 2 | 0 |
| 28 | 3 | 0 |
| 29 | 0 | 0 |
| 30 | 0 | 0 |
| 31 | 5 | 0 |
| 32 | 0 | 0 |
| 33 | 0 | 0 |
| 34 | 0 | 0 |
| 35 | 0 | 0 |
| 36 | 5 | 0 |
| 37 | 0 | 0 |
| 38 | 0 | 0 |
| 39 | 0 | 0 |
| 40 | 0 | 0 |

**Bước 6**:

N’ hiện tại là null vòng lặp bị dừng lại và tiếp tục B7

**Bước 7**:

- Thêm tập r vừa tìm được vào R.

* Tính trọng lượng của từng item trong P có thuộc tính thuộc tập r

t.weight ← α · t.weight (α = 1/3)

Các item P khi này bị giảm 2/3 trọng số cho mỗi item chứa thuộc tính {8}:

**Bảng 3.7. Xác định lại trọng số**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Số lần mang thai | Nồng độ glucose | Huyết áp (mm HG) | Độ dày da (mm) | Huyết thanh insulin(mu U/ml) | Chỉ số khối cơ thể (kg/m^2) | Chức năng phả hệ tiểu đường | Tuổi (năm) | Biến phân lớp |  |
| 1 | 8 | 13 | 17 | 21 | 27 | 31 | 36 | 41 | 0.33 |
| 1 | 8 | 13 | 17 | 21 | 28 | 31 | 36 | 41 | 0.33 |
| 2 | 8 | 14 | 16 | 21 | 27 | 31 | 36 | 41 | 0.33 |
| 3 | 8 | 11 | 16 | 21 | 28 | 31 | 36 | 41 | 0.33 |
| 2 | 8 | 14 | 16 | 21 | 28 | 31 | 36 | 41 | 0.33 |
| 3 | 9 | 14 | 16 | 21 | 28 | 33 | 39 | 41 | 1 |

Mảng PN’ khi này là:

**Bảng 3-8. PN sau khi trọng số giảm**

|  |  |  |
| --- | --- | --- |
| Value (a) | Trọng số của a trong P | Trọng số của a trong N |
| 1 | 0.67 | 4 |
| 2 | 0.67 | 2 |
| 3 | 1.33 | 3 |
| 4 | 0 | 0 |
| 5 | 0 | 0 |
| 6 | 0 | 0 |
| 7 | 0 | 1 |
| 8 | 1.67 | 0 |
| 9 | 1 | 3 |
| 10 | 0 | 5 |
| 11 | 0.33 | 0 |
| 12 | 0 | 1 |
| 13 | 0.67 | 6 |
| 14 | 1.67 | 2 |
| 15 | 0 | 0 |
| 16 | 2 | 4 |
| 17 | 0.67 | 4 |
| 18 | 0 | 1 |
| 19 | 0 | 0 |
| 20 | 0 | 0 |
| 21 | 2.67 | 6 |
| 22 | 0 | 1 |
| 23 | 0 | 0 |
| 24 | 0 | 1 |
| 25 | 0 | 1 |
| 26 | 0 | 0 |
| 27 | 0.67 | 2 |
| 28 | 4 | 5 |
| 29 | 0 | 1 |
| 30 | 0 | 0 |
| 31 | 5 | 5 |
| 32 | 0 | 3 |
| 33 | 1 | 0 |
| 34 | 0 | 0 |
| 35 | 0 | 1 |
| 36 | 1.67 | 2 |
| 37 | 0 | 2 |
| 38 | 0 | 4 |
| 39 | 1 | 1 |
| 40 | 0 | 0 |

Xét điều kiện của vòng lặp (1) để quyết định đi tiếp hoặc dừng lại.

* Tổng trọng của P bây giờ là 2.67 lớn hơn ngưỡng trọng số (0.3) -> vì vậy tiếp tục trong vòng lặp, quay trở lại Bước 3 ta có:

Trên vòng lặp tiếp theo này, thuộc tính với độ lợi gain lớn nhất bây giờ là 33 (gain = 0.92), vì vậy ta thêm cái này vào nguyên tắc R {33}-> {41}.

Qui trình cuối cùng ta sẽ nhận được 1 danh sách luật R:

(1) {8} 🡪 {41}

(2) {10} 🡪 {42}

(3) {38} 🡪 {42}

(4) {33} 🡪 {41}

**Bước 8**: Độ chính xác Laplace của luật được tính theo công thức sau:

**Laplace= (nc + 1) / (ntotal + k)**

Trong đó:

**nc**  : *tổng các mẫu tích cực của P ban đầu thỏa mãn quy luật tiền lệ r.*

**ntotal** : *tổng các mẫu (của P và N ban đầu ) thỏa mãn quy luật tiền lệ r.*

**k**  : *tổng số lớp được chia*

Ví dụ : Laplace (1) ((1)= 8🡪41) = (5+1)/(5+2)=0.86 (86%)

Tương tự như thế ta có kết quả:

(1) {8} 🡪 {41} 86%

(2) {10} 🡪 {42} 86%

(3) {38} 🡪 {42} 83%

(4) {33} 🡪 {41} 67%

Với độ laplace được sắp xếp giảm dần như sau:

8 ->41 độ tin cậy là: *86%*

Nồng độ glucose huyết tương 2 giờ trong một thử nghiệm dung nạp glucose đường uống =8 thì có 86% mắc bệnh.

10 -> 42 độ tin cậy là: *86%*

Nồng độ glucose huyết tương 2 giờ trong một thử nghiệm dung nạp glucose đường uống =10 thì có 86% không mắc bệnh.

38 -> 42 độ tin cậy là: *83%*

Tuổi =38 thì có 83% không mắc bệnh:

33 -> 41 độ tin cậy là: *67%*

Chức năng phả hệ tiểu đường =33 có 67% mắc bệnh.

### 3.5. PHÂN LỚP DỰA TRÊN LUẬT KẾT HỢP ĐÃ TÌM THẤY

- Mỗi tập luật chứa luật tương ứng với mỗi lớp, chúng ta sẽ sử dụng luật k tốt nhất của mỗi lớp cho dự đoán, với qui trình sau:

(1): Chọn tất cả các luật có giá trị thuộc item của itemset

(2): Từ tập luật được chọn ở bước (1), chọn luật tốt nhất của mỗi nhãn lớp (áp dụng cho trường hợp trùng luật mà sinh ra các độ chính xác khác nhau)

(3): Tính giá trị trung bình độ chính xác của k luật thuộc từng nhãn lớp và chọn lớp có độ chính xác cao nhất.

- Lý do chúng ta sử dụng qui tắc này là một luật đơn không thể phân lớp một cách hoàn hảo được. Chúng ta sử dụng nhiều qui tắc trong dự đoán vì độ chính xác của luật không thể được ước tính chính xác, và không thể mong đợi rằng 1 luật đơn có thể dự đoán hoàn hảo nhãn lớp của mỗi ví dụ trong tập dataset. Hơn nữa, chúng ta sử dụng k luật tốt nhất thay vì tất cả các luật vì số lượng luật khác nhau sẽ cho luật khác nhau, và chúng ta không muốn sử dụng xếp hạng luật khi đã có đủ quy tắc để đưa ra dự đoán. Giá trị k sau những lần thực nghiệm được lấy là 5.

(4): Với các tập luật được đưa ra, độ tin cậy được tính dựa trên số lần xuất hiện của thuộc tính trên kết quả nhãn lớp được tìm thấy ở bước (3).

**Ví dụ:**

Dựa vào ví dụ được đưa ra mục 4.3 ta có tập luật:

(1) Nồng độ glucose {8} 🡪 {41} 86%

(2) Huyết áp {10} 🡪 {42} 86%

(3) Tuổi {38} 🡪 {42} 83%

(4) Chức năng phả hệ tiểu đường {33} 🡪 {41} 67%

Phân lớp itemset dưới đây.

**Bảng 3-8. Phân lớp cho Itemset**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Số lần mang thai (1) | Nồng độ glucose (2) | Huyết áp (mm HG) (3) | Độ dày da (mm) (4) | Huyết thanh insulin(mu U/ml) (5) | Chỉ số khối cơ thể (kg/m^2) (6) | Chức năng phả hệ tiểu đường (7) | Tuổi (năm) (8) |
| 2 | 10 | 13 | 16 | 22 | 27 | 33 | 38 |

B1: Ta có tập luật với lỗi luật thuộc itemset:

(2) Huyết áp {10} 🡪 {42} 86%

(3) Tuổi {38} 🡪 {42} 83%

(4) Chức năng phả hệ tiểu đường {33} 🡪 {41} 67%

B2: Chọn luật có độ chính xác cao nhất

* Ở đây không có trường hợp bị trùng luật, nếu có luật có độ chính xác cao nhất sẽ được chọn

B3: Theo như trình bày ở trên ta phải lấy k=5 luật có độ chính xác cao nhất nhưng ở đây chỉ thu về 3 luật thỏa mãn phân lớp. Tính giá trị trung bình:

* Nhãn lớp 41: 67%
* Nhãn lớp 42: (86 + 83)/2 = 84%

Ta thấy nhãn lớp 41 có độ chính xác thấp hơn 42 nên ta kết luận itemset đã cho thuộc nhãn lớp 42.

B4: Tính độ tin cậy: Nhãn lớp 42: (1+1)/8=25%

## **Chương 4: ĐỀ XUẤT Ý TƯỞNG CẢI TIẾN.**

### 4.1. TRÌNH BÀY Ý TƯỞNG

Chính vì những vấn đề trên chúng em đề xuất việc không điền giá trị dự đoán nữa, mà ta dùng tập giá trị thiếu kia để tìm ra luật. Nếu thực hiện theo nghĩa đen nghĩa là không điền gì hết thì thuật toán sẽ không thể chạy như đã trình bày ở [chương 3.](#_Chương_3:_PHÂN)

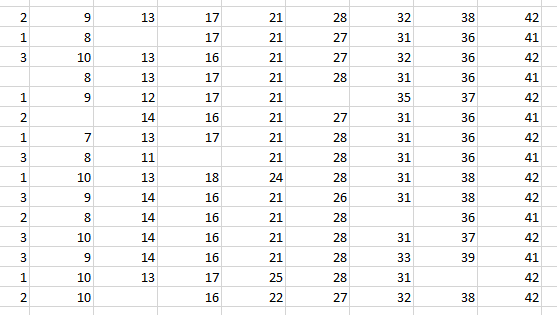
Trên thực tế chúng ta thu thập dữ liệu từ các câu hỏi khảo sát từ những người nghiên cứu sẽ mắc phải nhiều nguyên nhân chủ quan hay khách quan nên có những mục không được điền, những dữ liệu này được gọi là dữ liệu thô. Câu hỏi đặt ra là làm thế nào để xử lý dữ liệu thô này trước khi đưa vào phân tích nghiên cứu cho mục đích phân lớp. Nguyên nhân gây ra giá trị trống có thể so người trả lời không muốn trả lời, do bất cẩn nên bỏ qua một số câu hỏi hoặc do quá trình nhập liệu sai sót. Tỷ lệ giá trị trống sẽ làm giảm đi tính thuyết phục.

Ý tưởng cải tiến thuật toán ở đây sẽ là điền thêm các giá trị vào những vị trí trống, một số phương pháp điền giá trị trống được đưa ra:

* + Từ bỏ dòng bị thiếu dữ liệu.
  + Điền giá trị 0.
  + Điền giá trị đa số: Là giá trị có số lần xuất hiện trên cột nhiều nhất.
  + Điền giá trị liền kề: Lấy 2 giá trị bên trên và 2 giá trị bên dưới sau đó tính trung bình.

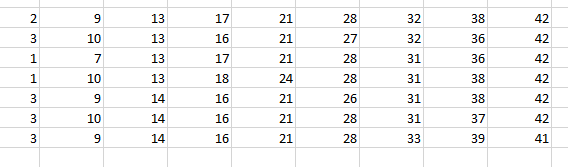
### 4.2. THỰC HIỆN Ý TƯỞNG

**Ví dụ:** Bộ dữ liệu đầu vào bị khuyết một số chỗ:



Hình 3.15. Ví dụ dữ liệu bị missing

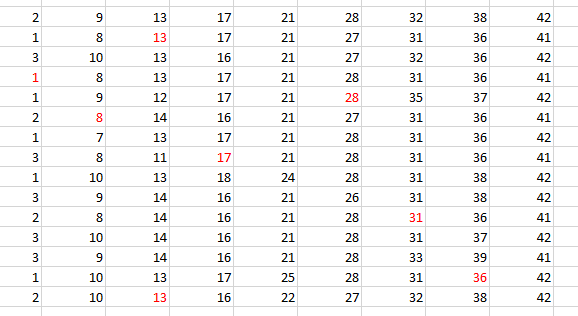
* Loại bỏ những dòng có giá trị trống:



Hình 3.16.Dữ liệu demo đã loại bỏ missing value

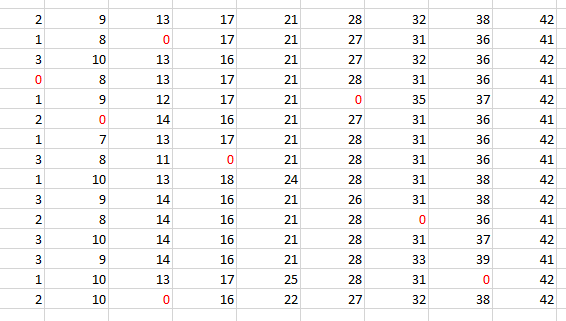
Tập dữ liệu đầu vào sẽ giảm, độ chính xác của thuật toán sẽ không cao.

* Điền giá trị xuất hiện nhiều nhất:



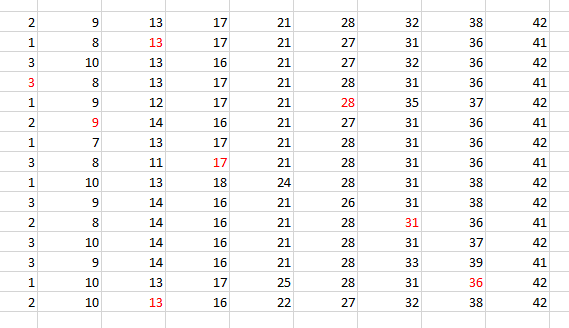
Hình 3.17.Điền giá trị xuất hiện nhiều nhất

* Điền số 0:



Hình 3.18.Điền giá trị 0

* Điền giá trị lân cận:



Hình 3.19.Điền giá trị lân cận

## **Chương 5: THỰC NGHIỆM**

### 5.1. MÔI TRƯỜNG THỰC NGHIỆM

Yêu cầu máy có cấu hình RAM 4g, CPU 2.5 GHz, core i3 trở lên. Ngôn ngữ lập trình là python 3.7, công cụ IDE Pycharm 2018.2.4.

Các thư viện sử dụng.

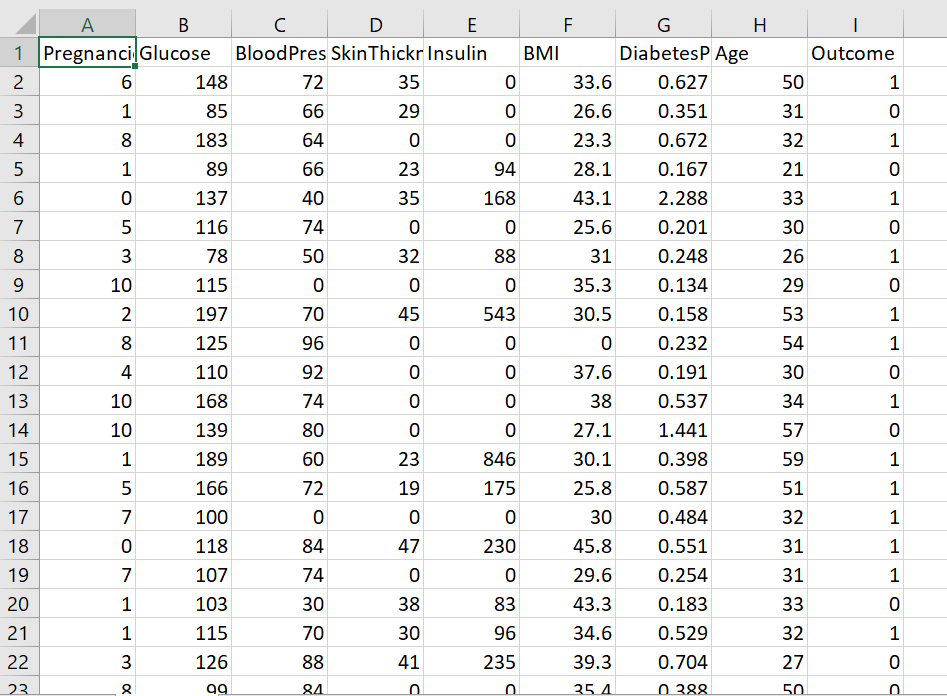
**Bảng 5‑1.Các thư viện sử dụng trong project**

|  |  |  |
| --- | --- | --- |
| Thư viện | Version | Chức năng |
| pySimpleGUI | 3.39 | Tạo giao diện |
| Xlrd | 1.2 | Xử lý file excel |
| Time | 1.0 | Đếm thời gian |
| Pandas | 0.23.4 | Xử lý dữ liệu dạng bảng |
| Math |  | Các hàm tính toán |
| Numpy | 1.16.2 | Xử lý dữ liệu dạng ma trận |
| Xlsxwriter | 1.1.8 | Viết ra file excel |
| random |  | Random số hàng cột dùng cho missing value |

### 5.2. DỮ LIỆU THỰC NGHIỆM

**5.2.1. Bệnh nhân tiểu đường ấn độ (Pima Indians)**

- **Mẫu dữ liệu**



**Hình 5.1.Mẫu dữ liệu pima indian**

- **Nguồn**: <https://www.kaggle.com/uciml/pima-indians-diabetes-database>

- **Thông tin đính kèm**: Bộ dữ liệu này có nguồn gốc từ Viện Tiểu đường và Bệnh tiêu hóa và Thận. Mục tiêu của bộ dữ liệu là dự đoán chẩn đoán xem bệnh nhân có bị tiểu đường hay không, dựa trên các phép đo chẩn đoán nhất định có trong tập dữ liệu. Một số ràng buộc đã được đặt vào việc lựa chọn các trường hợp này từ cơ sở dữ liệu lớn hơn. Đặc biệt, tất cả bệnh nhân ở đây là nữ ít nhất 21 tuổi của di sản Ấn Độ Pima.

- **Số mẫu tin**: 768 (mỗi mẫu tin là 1 dòng dữ liệu trong tệp).

- **Số lượng thuộc tính**: 9

- **Ý nghĩa của các thuộc tính**:

# Cột 1. Số lần mang thai (đơn vị là lần).

# Cột 2. Nồng độ glucose huyết tương 2 giờ trong một thử nghiệm dung nạp glucose đường uống.

# Cột 3. Huyết áp tâm trương (mm/Hg).

# Cột 4. Triceps da gấp độ dày (mm).

# Cột 5. 2-giờ huyết thanh insulin (mu U/ml).

# Cột 6. Chỉ số khối cơ thể (trọng lượng tính bằng kg/(chiều cao tính theo m)^2).

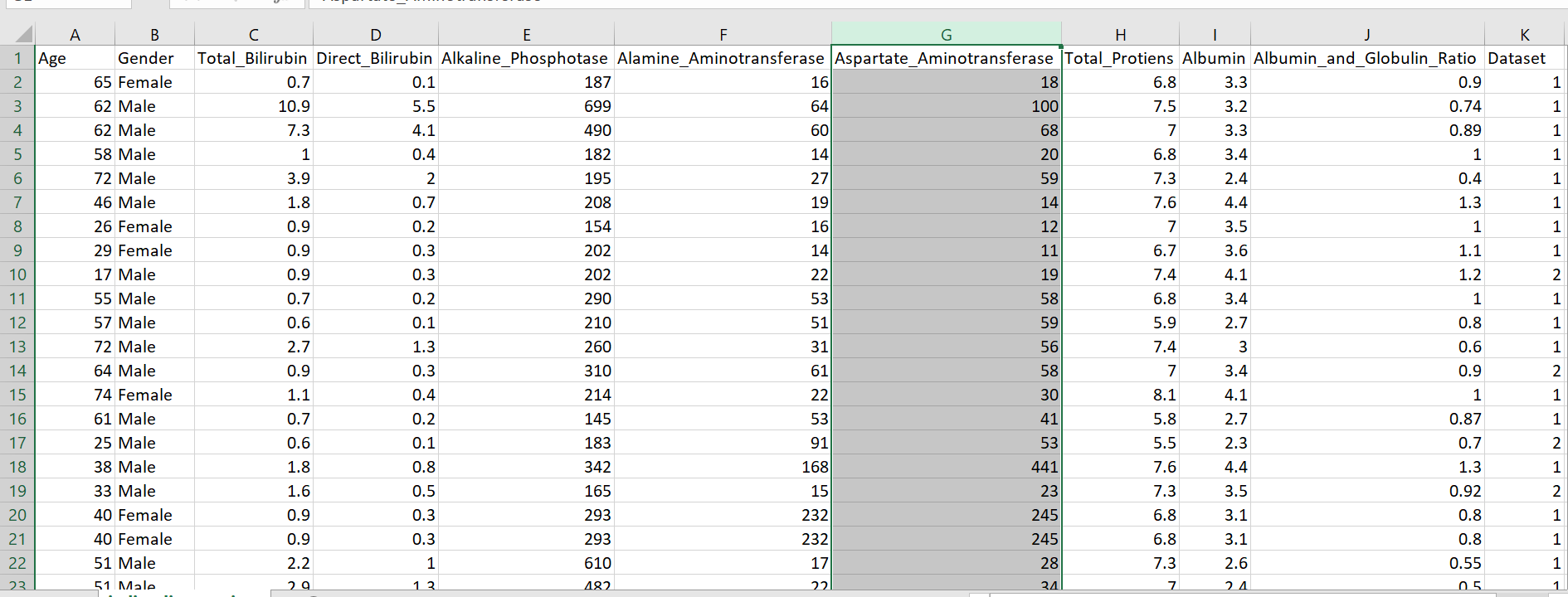
# Cột 7. Chức năng phả hệ tiểu đường.

# Cột 8. Tuổi (năm).

# Cột 9. Biến lớp (có mắc bệnh hoặc không mắc bệnh là 41, không mắc là 42)

* + 1. **Hồ sơ bệnh nhân gan Ấn Độ (Indian Liver Patient Records)**

**Mẫu dữ liệu**:



**Hình 5.2.Mẫu dữ liệu Indian Liver Patient Records**

**-Nguồn**: <https://www.kaggle.com/uciml/indian-liver-patient-records>

-**Thông tin đính kèm**: Bộ dữ liệu này chứa 416 hồ sơ bệnh nhân gan và 167 hồ sơ bệnh nhân không gan được thu thập từ Đông Bắc Andhra Pradesh, Ấn Độ. Cột "Bộ dữ liệu" là nhãn lớp được sử dụng để phân chia các nhóm thành bệnh nhân gan (bệnh gan) hoặc không (không có bệnh). Bộ dữ liệu này chứa 441 hồ sơ bệnh nhân nam và 142 hồ sơ bệnh nhân nữ.

Bất kỳ bệnh nhân nào có tuổi trên 89 được liệt kê là ở độ tuổi "90"

-**Số mẫu tin**: 583

-**Số lượng thuộc tính: 11**

-**Ý nghĩa các thuộc tính**.

#Cột 1. Age (Tuổi của bệnh nhân)

# Cột 2. Gender (Giới tính của bệnh nhân)

# Cột 3. Total\_Bilirubin (Tổng billirubin tính bằng mg / dL)

# Cột 4. Direct\_Bilirubin (Liên hợp billirubin tính bằng mg / dL)

# Cột 5. Alkaline\_Phosphotase (ALP tính bằng IU / L)

# Cột 6. Alamine\_Aminotransferase (ALT tính bằng IU / L)

# Cột 7. Aspartate\_Aminotransferase (AST tính theo IU / L)

# Cột 8. Total\_Protiens Total (Protein g / dL)

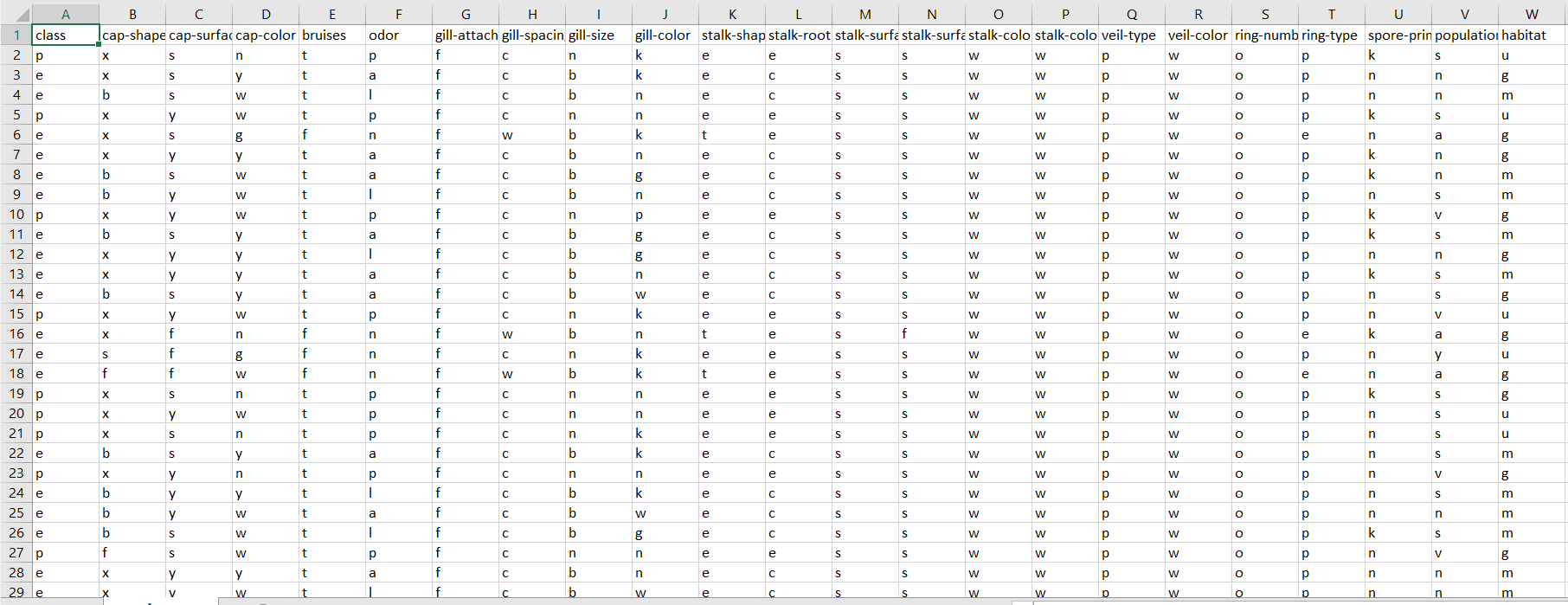
# Cột 9. Albumin (Albumin tính bằng g / dL)

# Cột 10. Albumin\_and\_Globulin\_Ratio (Tỷ lệ A / G)

# Cột 11. DatasetLabel (bệnh nhân có bị bệnh gan hay không)

* + 1. **Phân loại nấm (Mushroom Classification).**

-**Mẫu dữ liệu**:



**Hình 5.3.Mẫu dữ liệu Mushroom Classification**

-**Nguồn** <https://www.kaggle.com/uciml/mushroom-classification>

-**Thông tin đính kèm**: Bộ dữ liệu này bao gồm các mô tả về các mẫu giả thuyết tương ứng với 23 loài nấm mang trong “Nấm gia đình Agaricus và Lepiota” được rút ra từ “Hướng dẫn của Hội Audubon cho Nấm Bắc Mỹ (1981)”. Mỗi loài được xác định là chắc chắn có thể ăn được, chắc chắn là độc, hoặc không rõ khả năng ăn được và không được khuyến cáo. Lớp sau này được kết hợp với lớp độc. Hướng dẫn nêu rõ rằng không có quy tắc đơn giản nào để xác định tính ăn được của nấm.

-**Số lượng mẫu tin**: 8124

-**Số lượng thuộc tính**:23

-**Ý nghĩa các thuộc tính**:

# Cột 1. Class (thuộc lớp nào, có ăn được không): edible=e, poisonous=p

# Cột 2. Cap-shape (hình dáng đỉnh): bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s

# Cột 3. Cap-surface (bề mặt đỉnh): fibrous=f, grooves=g, scaly=y, smooth=s

# Cột 4. Cap-color (màu sắc đỉnh): brown=n, buff=b, cinnamon=c, gray=g, green=r, pink=p, purple=u, red=e, white=w, yellow=y

# Cột 5. Bruises (nó có bị bầm tím hay không): bruises=t, no=f

# Cột 6. Odor (mùi): almond=a, anise=l, creosote=c, fishy=y, foul=f, musty=m, none=n, pungent=p, spicy=s

# Cột 7. Gill-attachment (hình dáng): attached=a, descending=d, free=f, notched=n

# Cột 8. Gill-spacing (khoảng cách): close=c, crowded=w, distant=d

# Cột 9. Gill-size (độ rộng): broad=b, narrow=n

# Cột 10. Gill-color (màu sắc): black=k, brown=n, buff=b, chocolate=h, gray=g, green=r, orange=o, pink=p, purple=u, red=e, white=w, yellow=y

# Cột 11. Stalk-shape (hình dạng thân)

# Cột 12. Stalk-root (hình dạng gốc)

# Cột 13. Stalk-surface-above-ring (hình dáng bề mặt xung quanh đĩa trong)

# Cột 14. Stalk-surface-below-ring (hình dáng bề mặt bên dưới đĩa tròn)

# Cột 15. Stalk-color-above-ring (màu sắc xung quanh đĩa tròn)

# Cột 16. Stalk-color-below-ring (màu sắc bên dưới đĩa tròn)

# Cột 17. Veil-type (loại mạng che mặt)

# Cột 18. Veil-color (màu sắc màng che)

# Cột 19. Ring-number (số vòng)

# Cột 20. Ring-type (loại vòng)

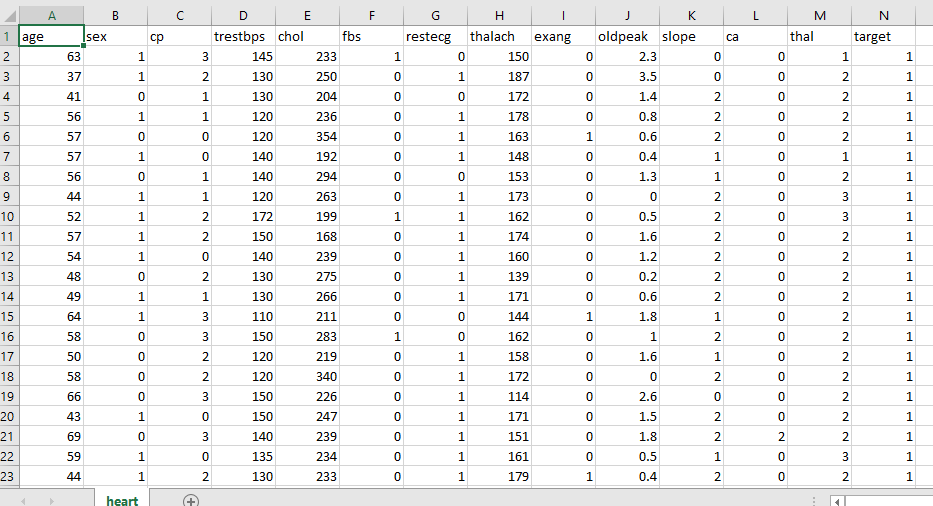
# Cột 21. Spore-print-color (bào tử, in màu)

# Cột 22. Population (số lượng cây)

# Cột 23. Habitat (môi trường sống).

#### 5.2.4. Phân loại bệnh tim

-**Mẫu dữ liệu**:



**Hình 5.3.Mẫu dữ liệu heart**

-**Nguồn** <https://www.kaggle.com/ronitf/heart-disease-uci>

-**Thông tin đính kèm**: Cơ sở dữ liệu này chứa 76 thuộc tính, nhưng tất cả các thử nghiệm được xuất bản đều đề cập đến việc sử dụng tập hợp con gồm 14 thuộc tính. Đặc biệt, cơ sở dữ liệu của Cleveland là cơ sở dữ liệu duy nhất được các nhà nghiên cứu ML sử dụng cho đến ngày nay. Trường "mục tiêu" đề cập đến sự hiện diện của bệnh tim ở bệnh nhân. Nó là số nguyên có giá trị từ 0 (không có sự hiện diện) đến 4.

-**Số lượng mẫu tin**: 4256

-**Số lượng thuộc tính**: 14

-**Ý nghĩa các thuộc tính**:

#Cột 1. Age: tuổi

#Cột 2. Sex: giới tính

#Cột 3. cp: loại đau ngực (4 giá trị)

#Cột 4. trestbps: huyết áp khi nghỉ ngơi

#Cột 5. chol: cholestoral huyết thanh tính bằng mg / dl

#Cột 6. Fbs: đường huyết lúc đói> 120 mg / dl

#Cột 7. Thalach: nghỉ ngơi kết quả điện tâm đồ (giá trị 0,1,2)

#Cột 8. Exang: nhịp tim tối đa đạt được

#Cột 9. Oldspeak: tập thể dục gây ra đau thắt ngực

#Cột 10. oldpeak = ST trầm cảm gây ra bởi tập thể dục liên quan đến nghỉ ngơi

#Cột 11. Slope: độ dốc của cuộc tập trận đỉnh cao đoạn ST

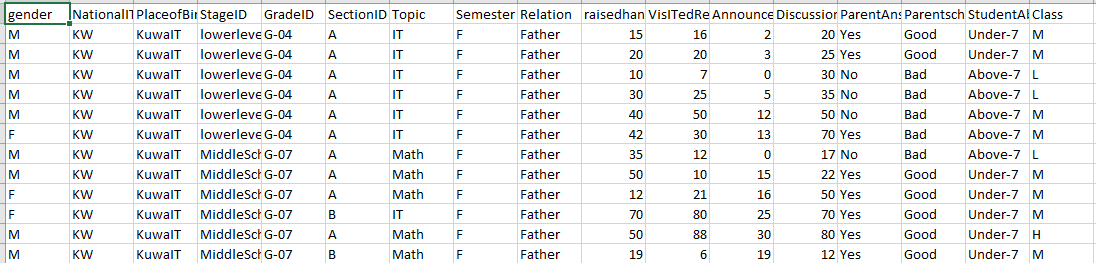
#Cột 12. số lượng tàu chính (0-3) được tô màu bằng flourosopy

#Cột 13. thal: 3 = bình thường; 6 = khuyết tật cố định; 7 = khuyết tật có thể đảo ngược

#Cột 14. Target: 1= bị bệnh tim, 0= không bị bênh tim

#### 5.2.5. Xếp loại sinh viên

-**Mẫu dữ liệu**:



**Hình 5.3.Mẫu dữ liệu Student**

- **Nguồn** <https://www.kaggle.com/ss2401/student-performance-in-class>

- **Thông tin đính kèm**: Dữ liệu phân loại học sinh dựa trên thông tin đời sống và kết quả học tập.

- **Số lượng mẫu tin**: 480

- **Số lượng thuộc tính**: 17

-**Ý nghĩa các thuộc tính**:

#Cột 1. Gender: giới tính

#Cột 2. NationalITy: Quốc tịch

#Cột 3. PlaceofBirth: Nơi sinh

#Cột 4. StageID: Giai đoạn

#Cột 5. GradeID: Hạng

#Cột 6. SectionID: Mục

#Cột 7. Topic: đề tài

#Cột 8. Semester: Học kỳ

#Cột 9. Relation: quan hệ

#Cột 10. Raisedhands: trưởng thành

#Cột 11. VisITedResources: nguồn tham khảo

#Cột 12. AnnouncementsView: thông báo

#Cột 13. Discussion: thảo luận

#Cột 14. ParentAnsweringSurvey: khảo sát phụ huynh

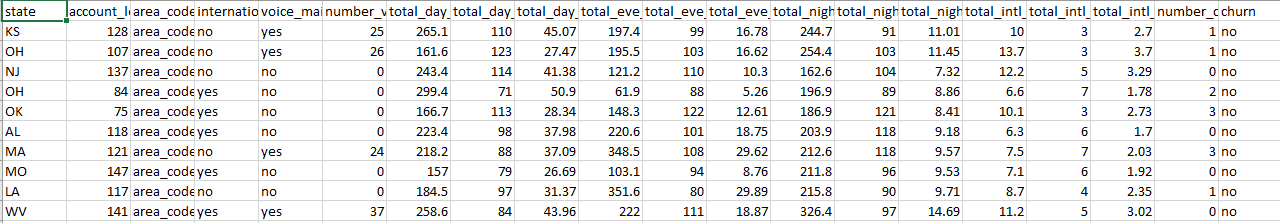
#Cột 15. ParentschoolSatisfaction: sự hài lòng của phụ huynh

#Cột 16. StudentAbsenceDays: ngày vắng mặt

#Cột 17. Class: phân loại

* + 1. **Tình trạng khách hàng (Churn)**

-**Mẫu dữ liệu**:



**Hình 5.3.Mẫu dữ liệu churn**

- **Nguồn** <https://www.kaggle.com/krleee/churn#churn.csv>

- **Thông tin đính kèm**: Dữ liệu chứa toàn bộ thông tin giao dịch của khách hàng được phân làm hai loại trung thành và không trung thành với ngân hàng.

- **Số lượng mẫu tin**: 5001

- **Số lượng thuộc tính**: 20

-**Ý nghĩa các thuộc tính**:

#Cột 1. state: tiểu bang

#Cột 2. account\_length: độ dài tài khoản

#Cột 3. area\_code: mã vùng

#Cột 4. International\_plan: chính sách quốc tế

#Cột 5. Voice\_mail\_plan

#Cột 6. Number\_vmail\_messages

#Cột 7. Total\_day\_minutes: Tổng số phút buổi sáng

#Cột 8. total\_day\_calls: Tổng cuộc gọi buổi sáng

#Cột 9. total\_day\_charge: Tổng phí buổi sáng

#Cột 10. total\_eve\_minutes: Tổng phút buổi chiều

#Cột 11. total\_eve\_calls: Tổng cuộc gọi buổi chiều

#Cột 12. total\_eve\_charge: Tổng phí buổi chiều

#Cột 13. total\_night\_minutes: Tổng phút buổi tối

#Cột 14. total\_night\_calls: Tổng cuộc gọi buổi tối

#Cột 15. total\_night\_charge: Chi phí cuộc gọi buổi tối

#Cột 16. total\_intl\_minutes: Tổng phút khuya

#Cột 17. total\_intl\_calls: Cuộc gọi khuya

#Cột 18. total\_intl\_charge: trả phí khuya

#Cột 19. number\_customer\_service\_calls: gọi dịch vụ chăm sóc khách hàng

#Cột 20. churn: yes là khách hàng thân thiết, no là khách hàng đã từ bỏ

### 5.3. QUY TRÌNH THỰC NGHIỆM VÀ TIÊU CHÍ ĐÁNH GIÁ

#### 5.3.1. Tạo tập dữ liệu có missing value

Để so sánh các phương pháp xử lý dữ liệu thiếu của tập dữ liệu đầu vào, chúng tôi sẽ chạy thực nghiệm với một vài tập dataset.

Để đảm bảo độ chính xác, chúng tôi sẽ chọn những tập dataset không có missing value, rồi tạo các tập dataset có missing value bằng cách loại bỏ ngẫu nhiên dữ liệu với các mức là 10%, 30%, 50%.

#### 5.3.2. Tiêu chí đánh giá.

Các tiêu chí đánh giá bao gồm:

* ĐỘ CHÍNH XÁC với đơn vị là %, tính bằng công thức DCX=T/N

Trong đó: T là tổng số phần tử được dự đoán chính xác nhờ luật.

N số lượng phần tử tập test.

* Tài nguyên sử dụng (CPU, RAM).
* Thời gian chạy (RUN TIME).

**5.3.3. Các bước thực nghiệm.**

Chạy các phương pháp xử lý missing value trên 4 tập như đã được trình bày ở mục 5.3.1 là tập hoàn hảo (tập ban đầu không có missing value), tập loại bỏ ngẫu nhiên 10% dữ liệu, tập loại bỏ ngẫu nhiên 30% và cuối cùng là 50% dữ liệu. Ở đây chúng tôi chạy các phương pháp xử lý missing value sau

1. Loại bỏ tất cả các dòng có missing value

(2) Các phương pháp điền dữ liệu thiếu như đã trình bày ở chương 3, [mục 3.4](#Các_PP_DIEN).

(3) Phương pháp được đề xuất, điền impossible value (điền chuỗi “null”).

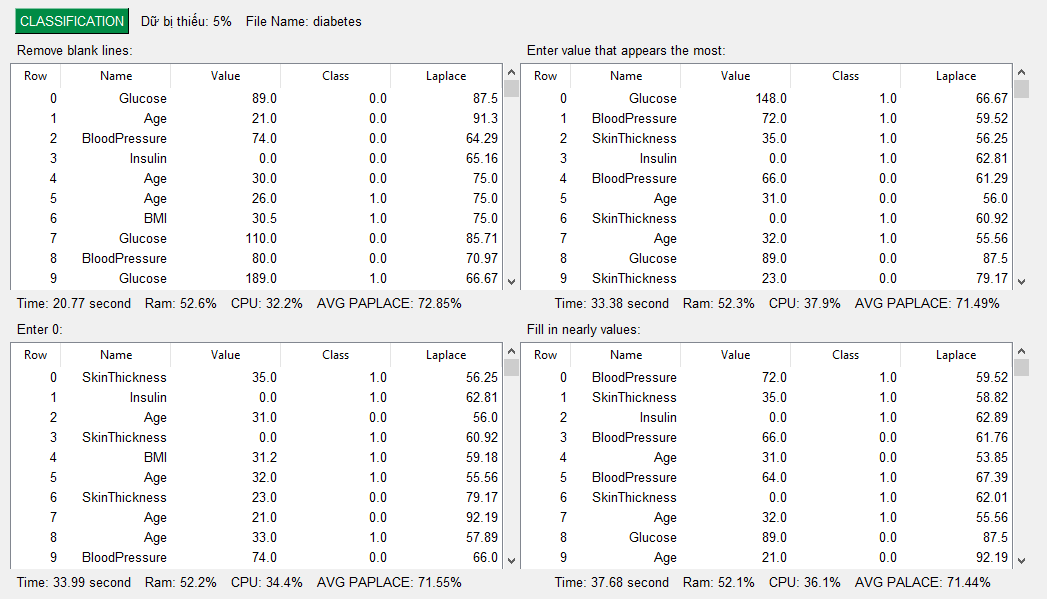
### 5.4. KẾT QUẢ CHẠY THỰC NGHIỆM

#### 5.4.1. Dataset: bệnh nhân tiểu đường (Pima Indians)

Bảng 5.2 trình bày kết quả thực nghiệm so sánh các phương pháp điền giá trị thiếu. Trường hợp thực nghiệm này, ta loại bỏ ngẫu nhiên mức 5% trên tổng số lượng các giá trị (không thuộc thuộc tính class) của tập dữ liệu gốc để tạo missing value. Số mising value = (số mẫu tin)\*(số thuộc tính[[1]](#footnote-1))\*0.05 = 768\*8\*0.05 = 307(giá trị).

**Bảng 5‑2.Các thông tin đánh giá thu được khi chạy với data:Tiểu đường khi loại bỏ ngẫu nhiên 5%**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PHƯƠNG PHÁP | ĐỘ CHÍNH XÁC (Đơn vị %) | CPU  (Đơn vị %) | RAM  (Đơn vị %) | RUN TIME  (mili giây) |
| Đã loại bỏ missing value | 72.85 | 32.2 | 52.6 | 20.77 |
| Thay missing=0 | 71.55 | 34.4 | 52.2 | 33.99 |
| Thay missing= giá trị đa số | 71.49 | 37.9 | 52.3 | 33.38 |
| Thay missing= giá trị lân cận | 71.44 | 36.1 | 52.1 | 37.68 |

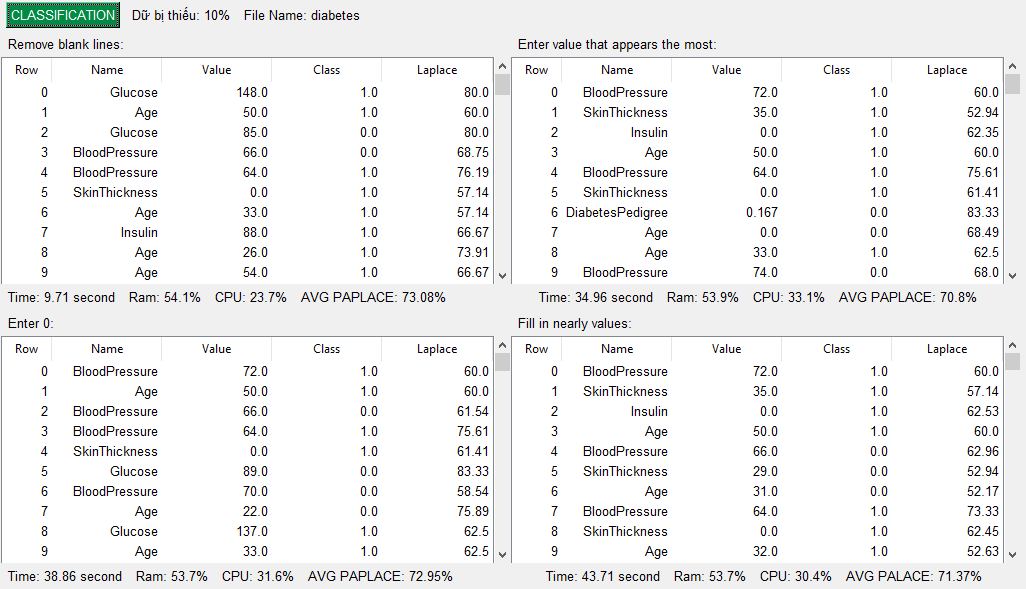


**Hình 5.4.Luật sinh ra khi tập Tiểu đường có missing value 5%**

Bảng 5.3 trình bày kết quả thực nghiệm so sánh các phương pháp điền giá trị thiếu. Trường hợp thực nghiệm này, ta loại bỏ ngẫu nhiên mức 10% trên tổng số lượng các giá trị (không thuộc thuộc tính class) của tập dữ liệu gốc để tạo missing value. Số mising value = (số mẫu tin)\*(số thuộc tính)\*0.1 = 768\*8\*0.1 = 614(giá trị).

**Bảng 5‑3. Các thông tin đánh giá thu được khi chạy với data:Tiểu đường khi loại bỏ ngẫu nhiên 10%**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PHƯƠNG PHÁP | ĐỘ CHÍNH XÁC (Đơn vị %) | CPU  (Đơn vị %) | RAM  (Đơn vị %) | RUN TIME  (mili giây) |
| Đã loại bỏ missing value | 73.08 | 23.7 | 54.1 | 9.71 |
| Thay missing=0 | 72.95 | 31.6 | 53.7 | 38.86 |
| Thay missing= giá trị đa số | 70.8 | 33.1 | 53.9 | 34.96 |
| Thay missing= giá trị lân cận | 71.37 | 30.4 | 53.7 | 43.71 |

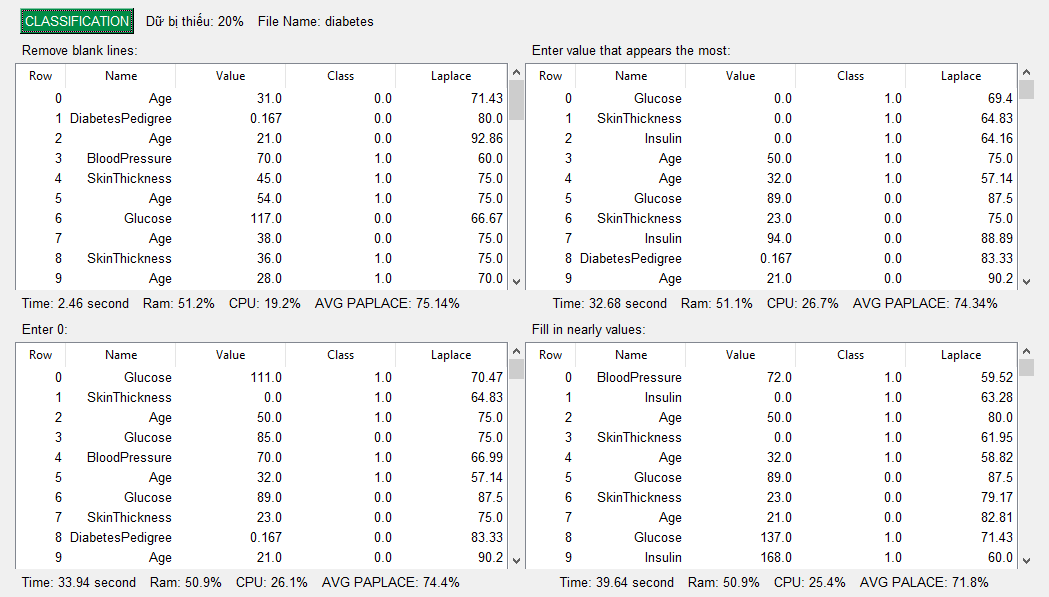


**Hình 5.5. Luật sinh ra khi tập Tiểu đường có missing value 10%**

Bảng 5.4 trình bày kết quả thực nghiệm so sánh các phương pháp điền giá trị thiếu. Trường hợp thực nghiệm này, ta loại bỏ ngẫu nhiên mức 20% trên tổng số lượng các giá trị (không thuộc thuộc tính class) của tập dữ liệu gốc để tạo missing value. Số mising value = (số mẫu tin)\*(số thuộc tính)\*0.2 = 768\*8\*0.2 = 1223(giá trị).

**Bảng 5‑4. Các thông tin đánh giá thu được khi chạy với data:Tiểu đường khi loại bỏ ngẫu nhiên 20%**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PHƯƠNG PHÁP | ĐỘ CHÍNH XÁC (Đơn vị %) | CPU  (Đơn vị %) | RAM  (Đơn vị %) | RUN TIME  (mili giây) |
| Đã loại bỏ missing value | 75.14 | 19.2 | 51.2 | 2.46 |
| Thay missing=0 | 74.4 | 26.1 | 50.9 | 33.94 |
| Thay missing= giá trị đa số | 74.34 | 26.7 | 51.1 | 32.68 |
| Thay missing= giá trị lân cận | 71.8 | 25.4 | 50.9 | 39.64 |



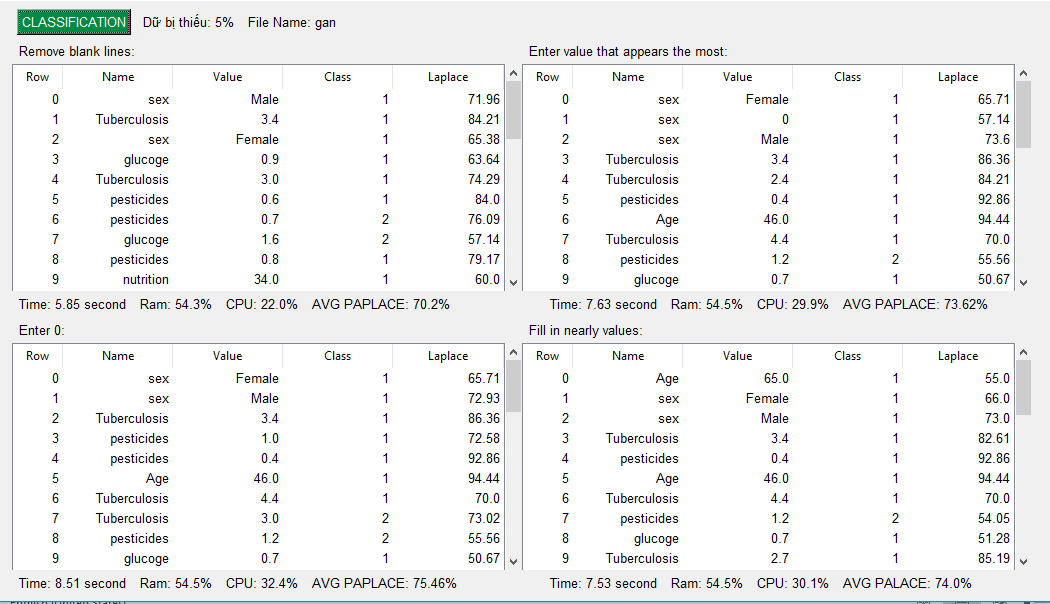
**Hình 5.6. Luật sinh ra khi tập Tiểu đường có missing value 20%**

#### 5.4.2. Dataset bệnh gan (Indian Liver Patient Records)

Bảng 5.5 trình bày kết quả thực nghiệm so sánh các phương pháp điền giá trị thiếu. Trường hợp thực nghiệm này, ta loại bỏ ngẫu nhiên mức 5% trên tổng số lượng các giá trị (không thuộc thuộc tính class) của tập dữ liệu gốc để tạo missing value. Số mising value = (số mẫu tin)\*(số thuộc tính)\*0.05 = 583\*10\*0.05 = 292(giá trị).

**Bảng 5‑5.Các thông tin đánh giá thu được khi chạy với data :Bệnh gan missing value 5%**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PHƯƠNG PHÁP | ĐỘ CHÍNH XÁC (ĐƠN VỊ %) | CPU  (Đơn vị %) | RAM  (Đơn vị %) | RUN TIME  (mili giây) |
| Đã loại bỏ missing value | 70.2 | 22.0 | 54.3 | 5.85 |
| Thay missing=0 | 75.46 | 32.4 | 54.5 | 8.51 |
| Thay missing= giá trị đa số | 73.62 | 29.9 | 54.5 | 7.63 |
| Thay missing= giá trị lân cận | 74.0 | 30.1 | 54.5 | 7.53 |

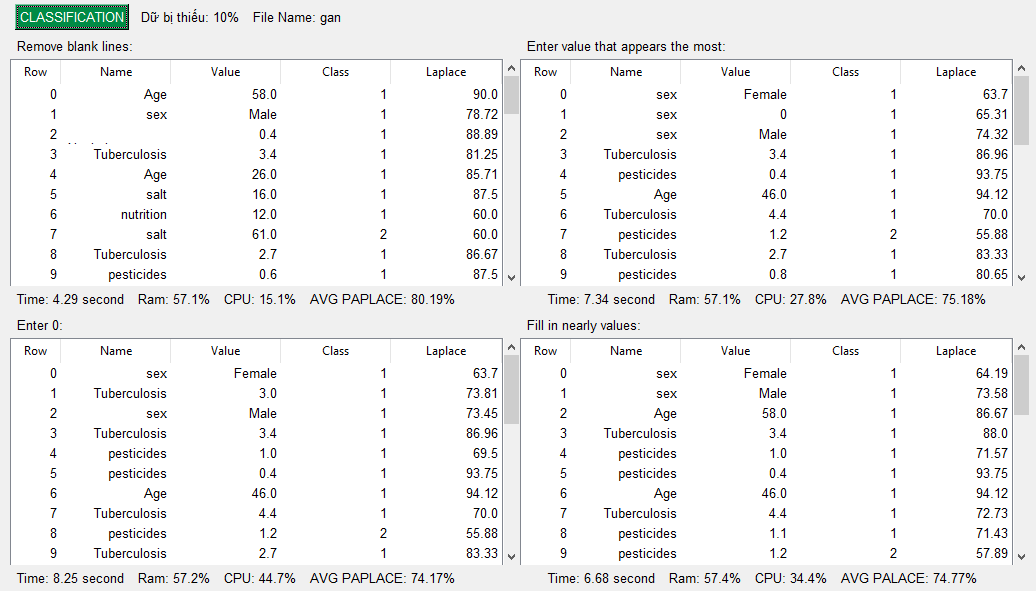


**Hình 5.7.Luật sinh ra khi tập Bệnh gan có missing value 5%**

Bảng 5‑8 trình bày kết quả thực nghiệm so sánh các phương pháp điền giá trị thiếu. Trường hợp thực nghiệm này, ta loại bỏ ngẫu nhiên mức 10% trên tổng số lượng các giá trị (không thuộc thuộc tính class) của tập dữ liệu gốc để tạo missing value. Số mising value = (số mẫu tin)\*(số thuộc tính)\*0.1 = 583\*10\*0.1 = 583(giá trị).

**Bảng 5-6.Các thông tin đánh giá thu được khi chạy với data :Bệnh gan missing value 10%**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PHƯƠNG PHÁP | ĐỘ CHÍNH XÁC (ĐƠN VỊ %) | CPU  (Đơn vị %) | RAM  (Đơn vị %) | RUN TIME  (mili giây) |
| Đã loại bỏ missing value | 80.19 | 15.1 | 57.1 | 4.29 |
| Thay missing=0 | 74.17 | 44.7 | 57.2 | 8.25 |
| Thay missing= giá trị đa số | 75.18 | 27.8 | 57.1 | 7.34 |
| Thay missing= giá trị lân cận | 74.77 | 34.4 | 57.4 | 6.68 |

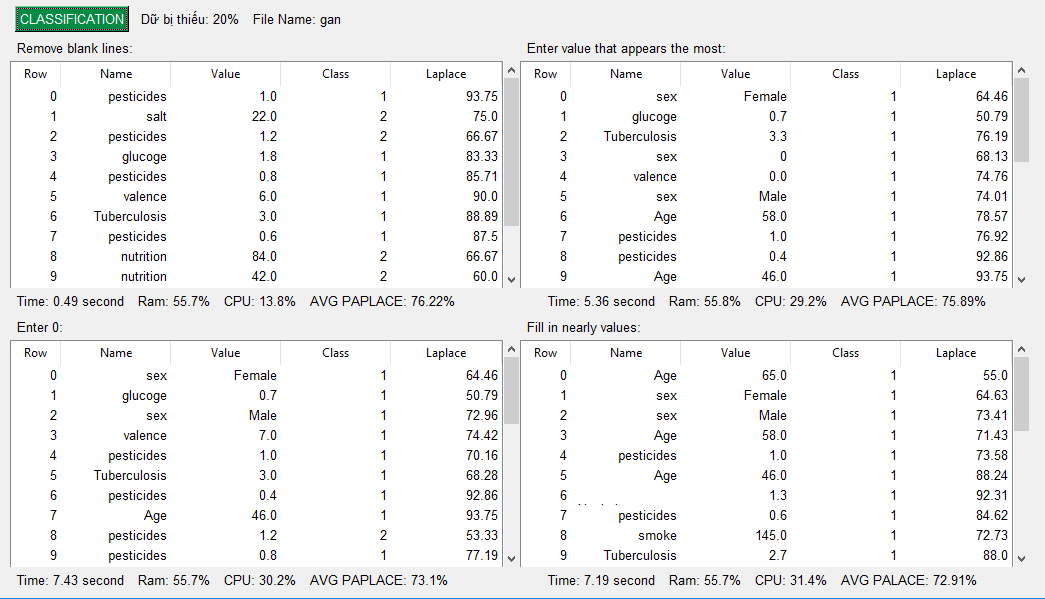


**Hình 5.8. Luật sinh ra khi tập Bệnh gan có missing value 10%**

Bảng 5.7 trình bày kết quả thực nghiệm so sánh các phương pháp điền giá trị thiếu. Trường hợp thực nghiệm này, ta loại bỏ ngẫu nhiên mức 20% trên tổng số lượng các giá trị (không thuộc thuộc tính class) của tập dữ liệu gốc để tạo missing value. Số mising value = (số mẫu tin)\*(số thuộc tính)\*0.2 = 583\*10\*0.2 = 1166(giá trị).

**Bảng 5‑7. Các thông tin đánh giá thu được khi chạy với data: Bệnh gan missing value 20%**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PHƯƠNG PHÁP | ĐỘ CHÍNH XÁC (ĐƠN VỊ %) | CPU  (Đơn vị %) | RAM  (Đơn vị %) | RUN TIME  (mili giây) |
| Đã loại bỏ missing value | 76.22 | 13.8 | 55.7 | 0.49 |
| Thay missing=0 | 73.1 | 30.2 | 55.7 | 7.43 |
| Thay missing= giá trị đa số | 75.89 | 29.2 | 55.8 | 5.36 |
| Thay missing= giá trị lân cận | 72.91 | 31.4 | 55.7 | 7.19 |



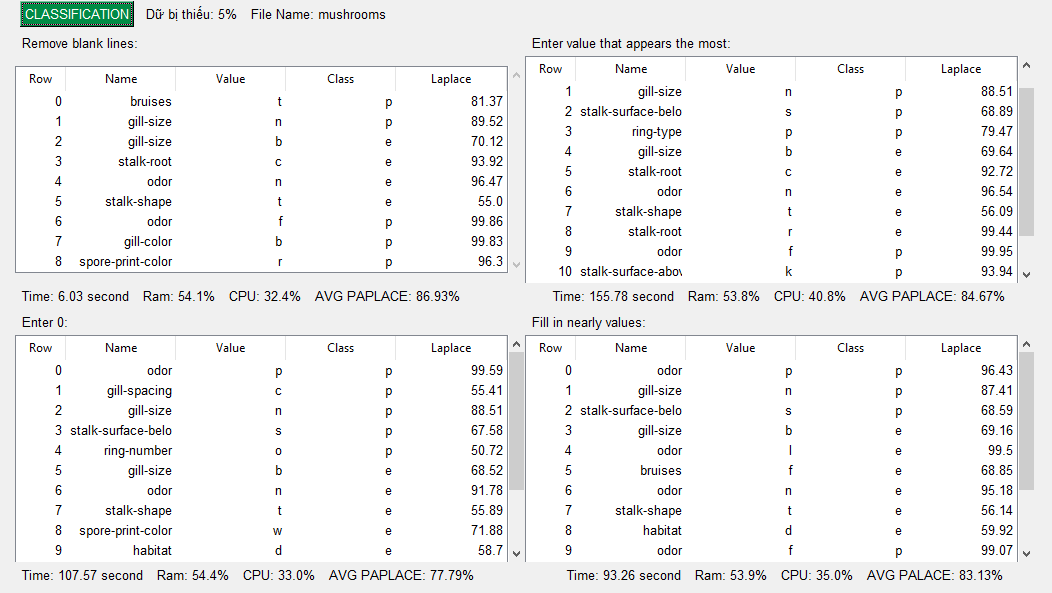
**Hình 5.9.Luật sinh ra khi tập Bệnh gan có missing value 20%**

#### 5.4.3. Dataset phân loại nấm (Mushroom Classification)

Bảng 5.8 trình bày kết quả thực nghiệm so sánh các phương pháp điền giá trị thiếu. Trường hợp thực nghiệm này, ta loại bỏ ngẫu nhiên mức 5% trên tổng số lượng các giá trị (không thuộc thuộc tính class) của tập dữ liệu gốc để tạo missing value. Số mising value = (số mẫu tin)\*(số thuộc tính)\*0.05 = 8124\*22\*0.05 = 8936(giá trị).

**Bảng 5‑8.Các thông tin đánh giá thu được khi chạy với data:Nấm missing value 5%**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PHƯƠNG PHÁP | ĐỘ CHÍNH XÁC (ĐƠN VỊ %) | CPU  (Đơn vị %) | RAM  (Đơn vị %) | RUN TIME  (mili giây) |
| Đã loại bỏ missing value | 96.93 | 32.4 | 54.1 | 6.03 |
| Thay missing=0 | 77.79 | 33.0 | 54.4 | 107.57 |
| Thay missing= giá trị đa số | 84.67 | 40.8 | 53.8 | 155.78 |
| Thay missing= giá trị lân cận | 83.13 | 35.0 | 53.9 | 83.26 |

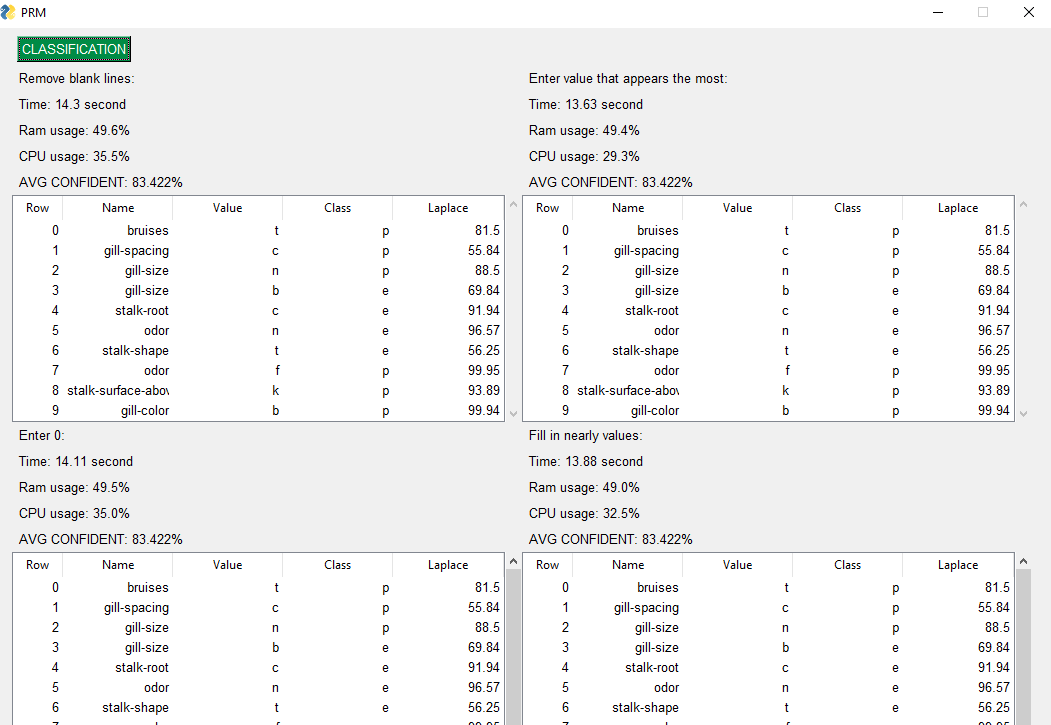


Hình 5.10.Luật sinh ra khi tập Dữ liệu Nấm có missing value 5%

Trường hợp thực nghiệm này, ta loại bỏ ngẫu nhiên mức 10% trên tổng số lượng các giá trị (không thuộc thuộc tính class) của tập dữ liệu gốc để tạo missing value. Số mising value = (số mẫu tin)\*(số thuộc tính)\*0.1 = 8124\*22\*0.1 = 17873(giá trị).

**Bảng 5‑9. Các thông tin đánh giá thu được khi chạy với data:Nấm missing value 10%**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PHƯƠNG PHÁP | ĐỘ CHÍNH XÁC (ĐƠN VỊ %) | | CPU  (Đơn vị %) | | RAM  (Đơn vị %) | RUN TIME  (mili giây) |
| Tập ban đầu | 83.42 | 32.45 | | 49.25 | | 13.9 |
| Đã loại bỏ missing value | 83.42 | 35.5 | | 49.6 | | 14.3 |
| Thay missing = 0 | 83.42 | 32.5 | | 49.0 | | 13.88 |
| Thay missing = giá trị đa số | 83.42 | 29.3 | | 49.4 | | 13.63 |
| Thay missing = giá trị lân cận | 83.42 | 32.5 | | 49.0 | | 13.88 |

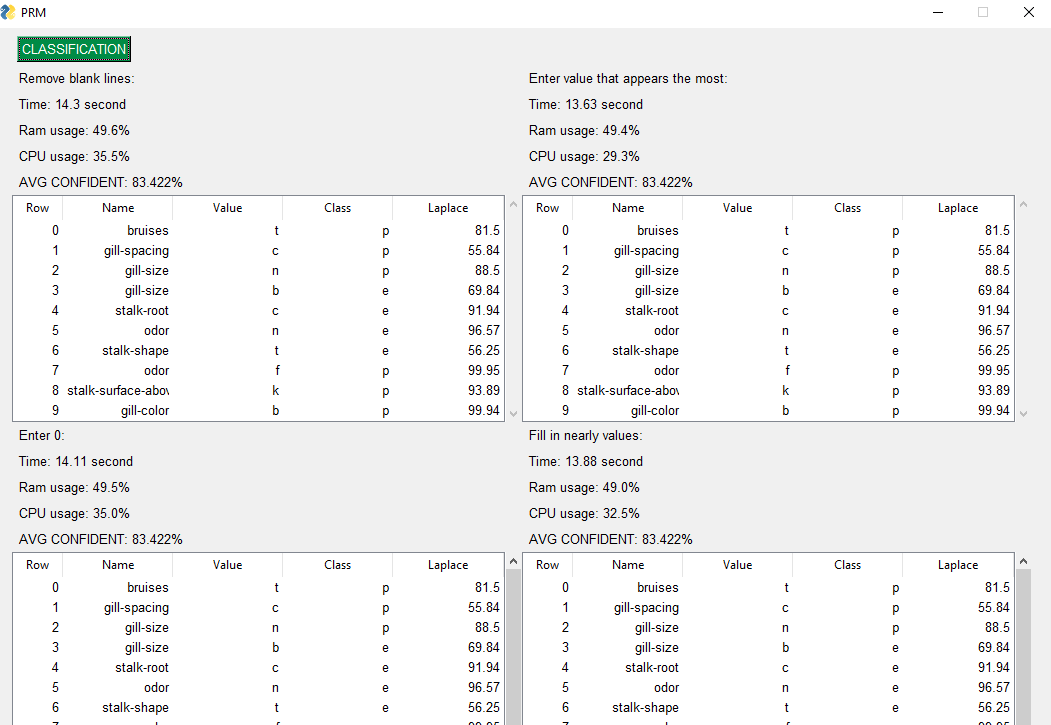


**Hình 5.11.Luật sinh ra khi tập Dữ liệu Nấm có missing value 10%**

Bảng 5.10 trình bày kết quả thực nghiệm so sánh các phương pháp điền giá trị thiếu. Trường hợp thực nghiệm này, ta loại bỏ ngẫu nhiên mức 20% trên tổng số lượng các giá trị (không thuộc thuộc tính class) của tập dữ liệu gốc để tạo missing value. Số mising value = (số mẫu tin)\*(số thuộc tính)\*0.2 = 8124\*22\*0.2 = 35746(giá trị).

**Bảng 5‑10. Các thông tin đánh giá thu được khi chạy với data: Nấm missing value 20%**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PHƯƠNG PHÁP | ĐỘ CHÍNH XÁC (ĐƠN VỊ %) | CPU  (Đơn vị %) | RAM  (Đơn vị %) | RUN TIME  (mili giây) |
| Tập ban đầu | 83.42 | 32.45 | 49.25 | 13.9 |
| Đã loại bỏ missing value | 83.42 | 35.5 | 49.6 | 14.3 |
| Thay missing=0 | 83.42 | 32.5 | 49.0 | 13.88 |
| Thay missing= giá trị đa số | 83.42 | 29.3 | 49.4 | 13.63 |
| Thay missing= giá trị lân cận | 83.42 | 32.5 | 49.0 | 13.88 |

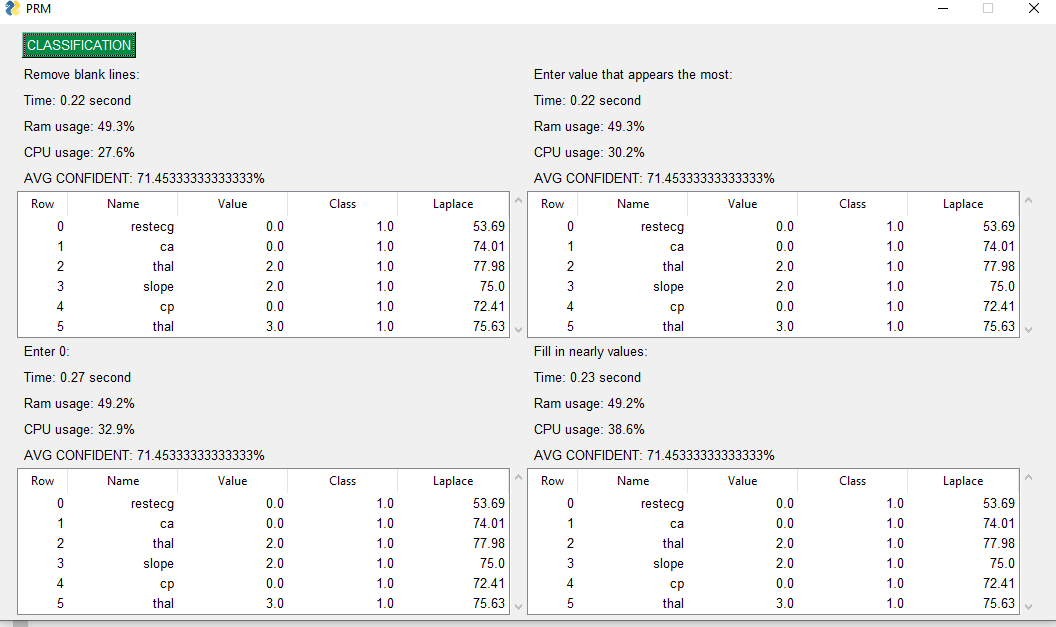


**Hình 5.12. Luật sinh ra khi tập Dữ liệu Nấm có missing value 20%**

#### 5.4.4. Dataset phân loại bệnh tim (Heart)

**Bảng 5‑8.Các thông tin đánh giá thu được khi chạy với data:Heart missing value**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PHƯƠNG PHÁP | ĐỘ CHÍNH XÁC (ĐƠN VỊ %) | CPU  (Đơn vị %) | RAM  (Đơn vị %) | RUN TIME  (mili giây) |
| Tập ban đầu | 83.42 | 32.45 | 49.25 | 13.9 |
| Đã loại bỏ missing value(1) | 71.45 | 27.6 | 49.3 | 0.22 |
| Thay missing=0 (2) | 71.45 | 30.2 | 49.3 | 0.22 |
| Thay missing= giá trị đa số (2) | 71.45 | 32.9 | 49.2 | 0.27 |
| Thay missing= giá trị lân cận (2) | 71.45 | 38.6 | 49.2 | 0.23 |



**Hình 5.10.Luật sinh ra khi tập Dữ liệu Heart có missing value**

**Phần 3: KẾT LUẬN**

**3.1. KẾT QUẢ SAU KHI HOÀN THÀNH BÁO CÁO**

Đề tài đã đáp ứng được yêu cầu mà giảng viên đề ra. Trong quá trình thực hiện có nhiều vấn đề phát sinh và gặp khó khăn tuy nhiên chúng em đã tập trung hết sức và nhận được sự giúp đỡ tận tình của người hướng dẫn thực tập. Đến nay đề tài đã hoàn thành và đạt được những kết quả cụ thể sau:

* Hiểu về thuật toán PRM, cũng như cách triển khai.
* Hiểu được kiến thức cơ bản về Python, cũng như công cụ PyCharm.
* Tạo project và thực thi thuật toán bằng ngôn ngữ Python trên PyCharm.
* Lên ý tưởng cải tiến thuật toán và thử nghiệm.

Đề tài khép lại mang đến lượng kiến thức và kinh nghiệm thực tế phong phú cho chính bản thân chúng em. Giúp chúng em có một cách làm khoa học, làm quen với môi trường làm việc đầy cạnh tranh và khắc nhiệt. Từ đó có một ý thực học tập cũng như làm việc nhóm, làm việc cá nhân thật tốt.

**3.2. KHÓ KHĂN GẶP PHẢI**

Nguồn tài liệu về đề tài còn khá ít nên quá trình tìm kiếm phục vụ cho đề tài gặp nhiều khó khăn.

Do kiến thức còn nhiều hạn chế và là ngôn ngữ mới đối với nhóm nên trong quá trình thực hiện gặp nhiều trục trặc và lỗi.

Do thời gian còn hạn hẹp và có nhiều đề tài nghiên cứu trong học kì nên chưa có cơ hội giới thiệu hết các chức năng nâng cao của PyCharm cũng như các lợi tích của ngôn ngữ bậc cao Python.

**3.3. ƯU ĐIỂM VÀ NHƯỢC ĐIỂM**

**Ưu điểm**

Giao diện đơn giản dễ sử dụng.

Đáp ứng đầy đủ các tính năng đưa ra ban đầu.

Đảm bảo chương trình hoạt động mượt mà, hạn chế tối đa lỗi có thể xảy ra.

Đảm bảo được độ chính xác của thuật toán khi bị mất dữ liệu ở mức độ nhất định

**Nhược điểm**

Chức năng chương trình khá đơn giản.

Tính năng còn nhiều hạn chế về việc tối ưu thuật toán cũng như với dữ liệu lớn.

Chương trình chưa được tối ưu hoàn toàn.

Những phương pháp cải tiến thuật toán đã đưa ra còn trong quá trình kiểm nghiệm.

**3.3. HƯỚNG PHÁT TRIỂN ĐỀ TÀI TRONG TƯƠNG LAI**

Hiện tại việc áp dụng ngôn ngữ Python đang là xu thế thịnh hành nên có nhiều cơ hội để phát triển trong tương lai.

Phát triển thêm đầy đủ tính năng của người dùng trong các lĩnh vực cần thiết.

Phân mềm có thể phân tích được mọi loại dữ liệu, tối ưu hóa thuật toán để tiết kiệm tài nguyên.

Bổ xung thêm một số phương pháp cải tiến thuật toán đáng tin cậy.

**DANH MỤC TÀI LIỆU THAM KHẢO**

[1] Jiawei Han, Micheline Kamber & Jian Pei, “DATA MINING”, Third Edition, 2012

[2] Bishop, Christopher M. “Pattern recognition and Machine Learning”, Springer (2016)

[3] [David G. Kleinbaum](https://www.google.com.vn/search?hl=vi&tbo=p&tbm=bks&q=inauthor:%22David+G.+Kleinbaum%22), [*Mitchel Klein*](https://www.google.com.vn/search?hl=vi&tbo=p&tbm=bks&q=inauthor:%22Mitchel+Klein%22)*-Logistic Regression: A Self-Learning Text*-Springer Science & Business Media, 14 thg 6, 2010.

[4] <https://cgi.csc.liv.ac.uk/~frans/KDD/Software/FOIL_PRM_CPAR/foil.html>

[5] Waseem Shahzad and Abdul Rauf Baig, “Hybrid associative classsification algorithm using ant colony optimization”. Department of Computer Science, National University of Computer and Emerging Sciences, Sector H-11/4, Islamabad, Pakistan. Received March 2010; revised July 2010

**[6]** Loan Nguyen, Bay Vo, and Tzung-Pei Hong, “CARIM: An Efficient Algorithm for Mining ClassAssociation Rules with Interestingness Measures”, Division of Knowledge and System Engineering for ICT, Ton Duc Thang University, Vietnam. Faculty of Information Technology, Ton Duc Thang University, Vietnam. Faculty of Information Technology, Ho Chi Minh City University of Technology, Vietnam. Department of CSIE, National University of Kaohsiung, Taiwan. Department of CSE, National Sun Yat-sen University, Taiwan. Vol. 12., No. 6A, 2015

[7] Jaber Alwidian, Bassam Hammo, and Nadim Obeid, “FCBA: Fast Classification Based on Association Rules Algorithm”, IJCSNS International Journal of Computer Science and Network Security, VOL.16 No.12, December 2016

[8] Jiawei Han University of Illinois at Urbana–Champaign Micheline Kamber Jian Pei Simon Fraser University*,* “2012- Data Mining. Concepts and Techniques, 3rd Edition”, ISBN 978-0-12-381479-1

[9] Frans Coenen, “THE LUCS-KDD IMPLEMENTATIONS OF CPAR (CLASSIFICATION BASED ON PREDICTIVE ASSOCIATION RULES)”, 16 February 2004.

[10] Waseem Shahzad and Abdul Rauf Baig, “HYBRID ASSOCIATIVE CLASSIFICATION ALGORITHM USING ANT COLONY OPTIMIZATION”, Article, November 2011, ISSN 1349-4198.

[11] Bijaya Kumar Nanda, Gyanesh Das, “Class Based Rule Mining using Ant Colony Optimization”, Department of ICT, F.M. University, Balasore, Odisha, department of ENTC, DRIEMS, ISSN (Print): 2319 – 2526, Volume-2, Issue-1, 2013

1. Số thuộc tính: là thuộc tính cần cho việc phân lớp - là thuộc tính có các giá trị riêng biệt và mỗi giá trị chỉ xuất hiện một lần, không bị trùng lặp. Ví dụ như: Pregnancies [↑](#footnote-ref-1)