**TRƯỜNG ĐẠI HỌC ĐIỆN LỰC**

**KHOA CÔNG NGHỆ THÔNG TIN**


# BÁO CÁO CHUYÊN ĐỀ HỌC PHẦN
# NHẬP MÔN HỌC MÁY

# ĐỀ TÀI: BÀI TOÁN PHÂN LỚP NHỊ PHÂN SỬ DỤNG SVM ĐỂ DỰ ĐOÁN KHẢ NĂNG SỐNG SÓT CỦA BỆNH NHÂN SUY TIM, KHẢ NĂNG MẮC BỆNH UNG THƯ VÚ


| | | |
|---|---|---|
| **Sinh viên thực hiện** | **:** | **NGUYỄN VĂN NAM** |
| **Giảng viên hướng dẫn** | **:** | **TS. NGÔ HOÀNG HUY** |
| **Ngành** | **:** | **CÔNG NGHỆ THÔNG TIN** |
| **Chuyên ngành** | **:** | **CÔNG NGHỆ PHẦN MỀM** |
| **Lớp** | **:** | **D13CNPM5** |
| **Khóa** | **:** | **2018-2023** |


**Hà Nội, tháng 06 năm 2021**

# PHIẾU CHẤM ĐIỂM

**Sinh viên thực hiện:**

| Họ và tên | Mã sinh viên | Nhiệm vụ | Chữ ký | Điểm |
|---|---|---|---|---|
| Nguyễn Văn Nam | 18810310428 | - Thu thập dữ liệu.<br>- Làm báo cáo.<br>- Viết code. | Nam | |

**Giảng viên chấm:**

| Họ và tên giảng viên | Nhận xét | Chữ Ký |
|---|---|---|
| Giảng viên chấm 1 : | | |
| Giảng viên chấm 2 : | | |

# MỤC LỤC

Contents

CHƯƠNG 1: TỔNG QUAN VỀ HỌC MÁY VÀ PHÂN LỚP NHỊ PHÂN VỚI MÔ HÌNH SVM ..... 1
1.1. Tổng quan về học máy ..... 1
1.1.1. Khái niệm học máy ..... 1
1.1.2. Các phương pháp học máy ..... 2
1.1.3. Ứng dụng của học máy ..... 4
1.2. Phân lớp nhị phân và mô hình SVM ..... 5
1.2.1. Phân lớp nhị phân ..... 5
1.2.2. Mô hình SVM ..... 6
1.2.2.1. Giới thiệu ..... 6
1.2.2.2. Support vector classifier phân lớp nhị phân với SVC ..... 7
CHƯƠNG 2: ỨNG DỤNG CỦA MÔ HÌNH SVM TRONG PHÂN LỚP NHỊ PHÂN ..... 9
2.1. Mô tả bài toán ..... 9
2.1.1. Bài toán dự đoán khả năng sống sót của bệnh nhân suy tim ..... 9
2.1.1.1. Mô tả ..... 9
2.1.1.2. Yêu cầu bài toán ..... 9
2.1.2. Bài toán dự đoán khả năng mắc bệnh ung thư vú ..... 9
2.1.2.1. Mô tả ..... 9
2.1.2.2. Yêu cầu bài toán ..... 10
2.2. Môi trường thực nghiệm ..... 10
2.2.1. Giới thiệu về ngôn ngữ python ..... 10
2.2.2. Cài đặt ngôn ngữ python và bộ thư viện cho bài toán ..... 11
2.3. Xây dựng bộ dữ liệu ..... 12
2.3.1. Bộ dữ liệu cho bài toán dự đoán khả năng sống sót của bệnh nhân suy tim ..... 12
2.3.2. Bộ dữ liệu cho bài toán dự đoán khả năng mắc bệnh ung thư vú ..... 13
2.4. Áp dụng thuật toán SVM vào bài toán dự đoán khả năng sống sót của bệnh nhân suy tim ..... 14
2.4.1. Cài đặt thuật toán ..... 14
2.4.2. Kết quả thực nghiệm ..... 16
2.5. Áp dụng thuật toán SVM vào bài toán dự đoán khả năng mắc bệnh ung thư vú ..... 21
2.5.1. Cài đặt thuật toán ..... 21

2.5.2. Kết quả thực nghiệm.....23
TÀI LIỆU THAM KHẢO .....29

# DANH MỤC HÌNH ẢNH

## Contents

Hình 1.1 Hình ảnh minh hoạ về học máy ..... 1
Hình 1.2: Bộ dữ liệu MNIST cho bài toán nhận dạng chữ viết tay ..... 2
Hình 1.3: Ứng dụng của phương pháp học không giám sát ..... 2
Hình 1.4: học bán giám sát ..... 3
Hình 1.5: ví dụ về học tăng cường ..... 3
Hình 1.6: Cảnh báo giao thông (trên ứng dụng Google Maps) ..... 4
Hình 1.7: Đề xuất gắn thẻ, nhận dạng của Machine Learning ..... 4
Hình 1.8: Minh hoạ phân tách tuyến tính ..... 7
Hình 1.9: Magin trong SVM ..... 8
Hình 2.1: Ngôn ngữ python ..... 10
Hình 2.2: Cài đặt thành công python ..... 11
Hình 2.3: Cài đặt thư viện numpy ..... 11
Hình 2.4: Cài đặt thư viện pandas ..... 11
Hình 2.5: Cài đặt thư viện sklearn ..... 11

# LỜI MỞ ĐẦU

Công nghệ ngày càng đạt được những thành tựu to lớn, đóng góp vai trò quan trọng trong đời sống của con người. “Machine Learning” một cụm từ được mọi người chú ý đến với khả năng tự học hỏi một cách nhanh chóng khi có một lượng dữ liệu nhất định cùng với thuật toán phù hợp.

Hiện nay, càng nhiều người chú ý và quan tâm đến Machine Learning bởi những thành tích đáng kể mà nó đem lại cho con người, ví dụ như giúp con người gia tăng dung lượng lưu trữ các loại dữ liệu sẵn, việc xử lý tính toán có chi phí thấp và hiệu quả hơn rất nhiều lần mang lại độ chính xác cao. Chính sự hiệu quả trong công việc và các lợi ích mà Machine Learning đem lại cho con người vì vậy em quyết định chọn đề tài: “Bài toán phân lớp nhị phân sử dụng SVM để dự đoán khả năng sống sót của bệnh nhân suy tim, khả năng mắc bệnh ung thư vú”.

# LỜI CẢM ƠN

Em xin chân thành cảm ơn giảng viên **TS. Ngô Hoàng Huy** đã truyền đạt cho em những kiến thức bổ ích, cần thiết, đầy đủ về học phần "Nhập môn học máy" trong những buổi học trực tiếp trên giảng đường, các buổi online cũng như thông qua quá trình trao đổi để em có thể tự tin hoàn thành tốt bài báo cáo chuyên đề của mình.

Đồng thời, thông qua việc tìm hiểu trên các trang mạng như google, các group trên facebook đã giúp em hiểu sâu hơn vấn đề mình cần giải quyết để giúp em có thể hoành thành tốt báo cáo chuyên đề với tất cả sự nỗ lực. Bên cạnh đó, với thời gian có hẹp cũng như sự hiểu biết không được sâu sắc, nhạy bén thì báo cáo chuyên đề của em không tránh khỏi được những thiếu xót. Em rất mong được sự đóng góp, chỉ dạy từ thầy cô để có thể có một báo cáo chuyên đề hoàn thiện nhất.

Sau cùng, em xin kính chúc thầy cô **Khoa Công Nghệ Thông Tin** nói chung, cũng như giảng viên **TS. Ngô Hoàng Huy** nói riêng có thật nhiều sức khoẻ để tiếp tục thực hiện sứ mệnh cao đẹp của những người thầy, người cô truyền đạt cho thế sau này những điều hay, ý đẹp.

Một lần nữa, em xin chân thành cảm ơn!

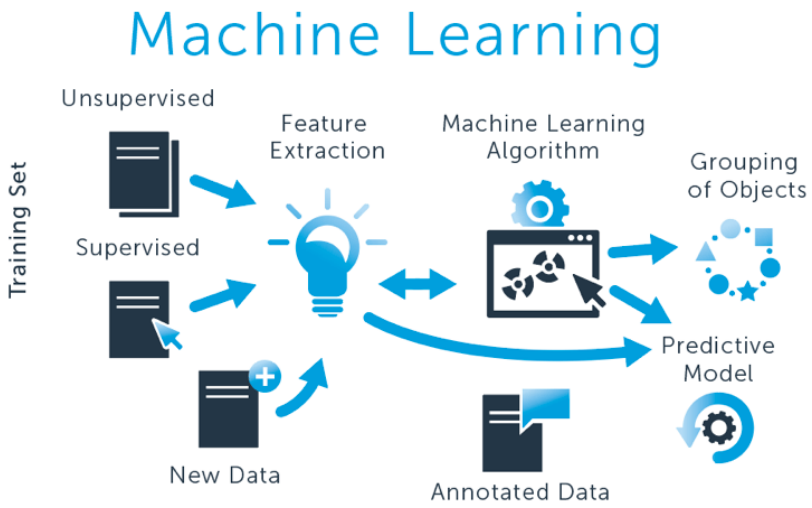
**Người thực hiện**

Nguyễn Văn Nam

# CHƯƠNG 1: TỔNG QUAN VỀ HỌC MÁY VÀ PHÂN LỚP NHỊ PHÂN VỚI MÔ HÌNH SVM

## 1.1. Tổng quan về học máy

### 1.1.1. Khái niệm học máy




Hình 1.1: Hình ảnh minh hoạ về học máy

Học máy hay máy học (tiếng anh: Machine Learning) là một thuật ngữ đề cập đến các chương trình máy tính có khả năng học hỏi về cách hoàn thành các nhiệm vụ và cải thiện hiệu suất theo thời gian.

Học máy là một công nghệ được phát triển từ lĩnh vực trí tuệ nhân tạo. Nó đòi hỏi sự đánh giá của con người trong việc tìm hiểu dữ liệu cơ sở và kĩ thuật phù hợp để phân tích dữ liệu. Đồng thời, trước khi sử dụng dữ liệu phải sạch, không có sai lệch cũng như không có dữ liệu giả.

Các mô hình Machine Learning yêu cầu dữ liệu đủ lớn để có thể huấn luyện và đánh giá mô hình. Trước đây, các thuật toán Machine Learning thiếu quyền truy cập vào một lượng lớn dữ liệu cần thiết để mô hình hoá các mỗi quan hệ giữa các dữ liệu. Sự tăng trưởng trong big data đã cung cấp các thuật toán Machine Learning với đủ dữ liệu cải thiện độ chính xác của mô hình học máy và dự đoán.

Những năm gần đây, khi mà khả năng tính toán của các máy tính được nâng lên một tầm cao mới và lượng dữ liệu khổng lồ được thu thập bởi các hãng công nghệ lớn, Machine Learning đã tiến thêm một bước dài và một lĩnh vực mới được ra đời gọi là Deep Learning. Lĩnh vực mới này giúp cho máy tính thực thi những việc tưởng chừng như không thể thực hiện vào 10 năm trước như phân loại cả ngàn vật thể khác nhau trong một bức ảnh, tự tạo chú thích cho ảnh, bắt chước giọng nói và chữ viết của con người, giao tiếp với con người, ….

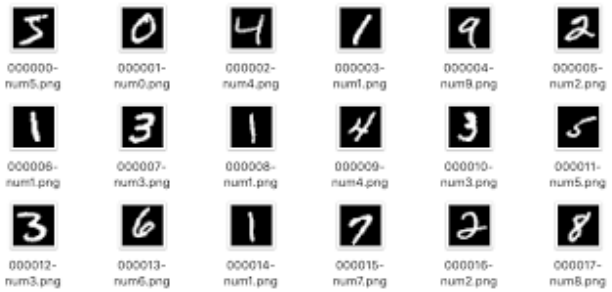
### 1.1.2. Các phương pháp học máy

#### 1.1.2.1. Học có giám sát (Supervised Learning)

Học có giám sát là thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới (new input) dựa trên cập (input, outcome) biết trước. Cặp dữ liệu này còn được gọi là (data, lable) tức là đầu vào sẽ là dữ liệu data cùng với nhãn của dữ liệu đó.

Tức là ta có một tập hợp biến đầu vào x = {x1, x2, x3, …, xn} và một tập nhãn tương ứng là y = {y1, y2, y3, …, yn} thì tập dữ liệu này được gọi là tập đữ liệu huấn luyện.

Ví dụ: trong bài toán nhận dạng chữ viết tay sử dụng bộ dữ liệu MNIST, ta có rất nhiều ảnh về các chữ số, thì để hiểu học có giám sát là gì thì ta có thể thấy rằng đầu vào của bài toán là những hình ảnh của các chữ số (hình ảnh này được gọi là data) kèm theo đó là nhãn của các hình ảnh (ví dụ hình ảnh là số 1 thì sẽ được đánh nhãn là "1"). Khi đó đầu ra sẽ trả lời cho câu hỏi hình ảnh đó là số mấy? có nghĩa là đầu ra là ta đưa vào một hình ảnh mới không có trong tập dữ liệu huấn luyện và trả lời đó là số mấy?

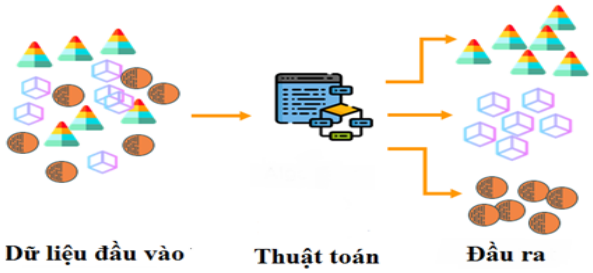


Hình 1.2. Bộ dữ liệu MNIST cho bài toán nhận dạng chữ viết tay

#### 1.1.2.2. Học không giám sát (UnSupervised Learning)

Đối với thuật toán học không giám sát này thì chúng ta không biết được đầu ra tức là nhãn của bài toán mà chỉ có dữ liệu đầu vào. Hay nói một cách khác thuật toán học không giám sát là chúng ta chỉ có dữ liệu đầu vào X mà không biết nhãn tương ứng Y.

Một ví dụ đơn giản, giống như việc không có thầy cô giáo nào dạy cho chúng ta biết chữ cái đó là chữ A hay chữ B, … \. Ứng dụng chủ yếu của bài toán học không giám sát là bài toán về phân cụm



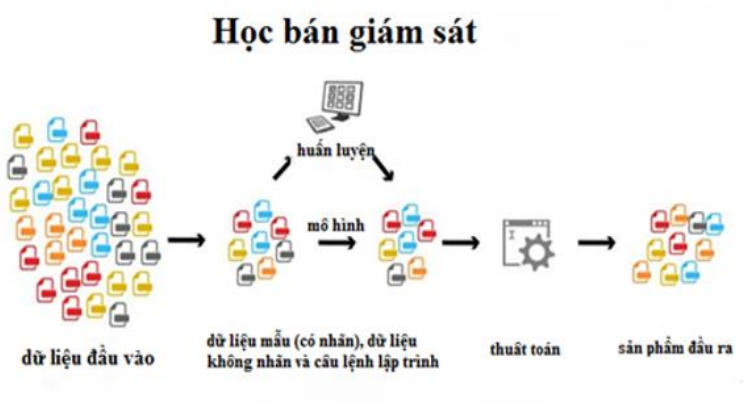

Hình 1.3: Ứng dụng của bài toán học không giám sát

### 1.1.2.3. Học bán giám sát (Semi- Supervised Learning)

Các bài toán khi chúng ta một lượng dữ liệu lớn X nhưng chỉ một phần trong tập dữ liệu đó được gán nhãn thì được gọi là học bán giám sát. Những bài toán thuộc nhóm này nămg giữa hai nhóm là học có giám sát và học không giám sát.

Một ví dụ điển hình cho học bán giám sát là chỉ có một phần ảnh hoặc văn bản được gán nhãn (ví dụ bức ảnh về người, về động vật hoặc các văn bản khoa học, …) và một phần lớn các bức ảnh/ văn bản đó chưa được gán nhãn được thu thập từ internet.

Thực tế có rất nhiều bài toán Machine Learning thuộc vào nhóm này vì việc thu thập dữ liệu có nhãn tốt rất nhiều thời gian và chi phí cao. Rất nhiều loại dữ liệu thậm chí cần phải có chuyên gia mới gán nhãn được, ngược lại dữ liệu chưa có nhãn có thể thu thập với chi phí thấp từ internet.

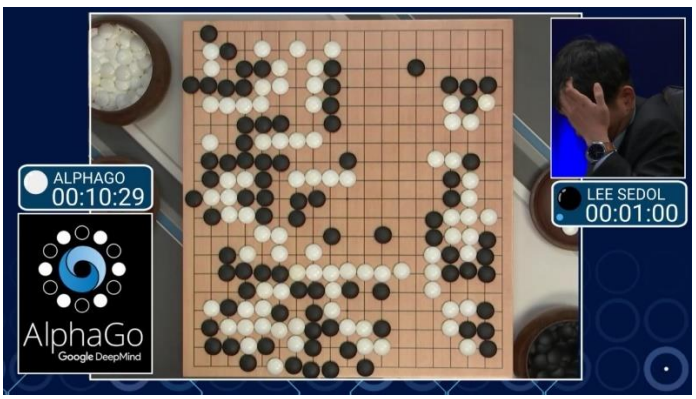



Hình 1.4: Học bán giám sát

### 1.1.2.4. Học tăng cường (Reindorcement Learning)

Học tăng cường là các bài toán giúp cho hệ thống tự động xác định được hành vi dựa trên hoàn cảnh để đạt được lợi ích cao nhất. Hiện tại học tăng cường chủ yếu được áp dụng vào lý thuyết trò chơi, các thuật toán cần xác định nước đi tiếp theo để đạt được điểm số cao nhất.

Ví dụ điển hình cho phương pháp học tăng cường này là: AlphaGo gần đây nổi tiếng với việc chơi cờ vây thắng cả con người.




Hình 1.5: Ví dụ về phương pháp học tăng cường

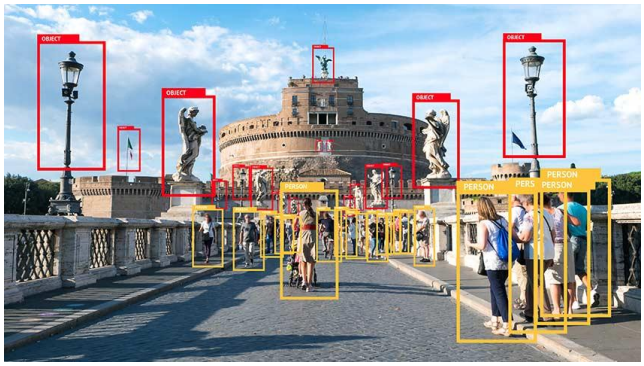
### 1.1.3. Ứng dụng của học máy

Học máy có rất nhiều ứng dụng tốt giúp cho con người thuận tiện hơn ví dụ như:

**Cảnh báo giao thông trên ứng dụng Google Maps**: giờ đây có lẽ ứng dụng được sử dụng với tần suất nhiều nhất mỗi khi bạn tham gia giao thông. Đặc biệt là khi các ứng dụng khác về di chuyển như Grab được áp dụng rộng rãi đồng nghĩa google maps được sử dụng liên tục để chỉ đường cho nhà cung cấp dịch vụ hay người sử dụng dịch vụ. Những thông tin về quãng đường tối ưu, thời gian di chuyển nhanh nhất cũng được phân tích cùng lúc trên google maps. Thực tế, dữ liệu của tuyến đường đó được thu thập theo thời gian và một số dữ liệu có từ nguồn khác. Mọi người sử dụng bản đồ đều cung cấp vị trí, tốc độ trung bình, tuyến đường. Những thông tin này được google thu thập và tổng hợp thành lưu lượng truy cập thông qua các thuật toán phân tích phức tạp trên Machine Learning, những thông tin này trở lên có nghĩa chúng giúp Google dự đoán lưu lượng sắp tới và điều chỉnh tuyến đường của bạn theo cách tối ưu nhất.

Hình 1.6: Cảnh báo giao thông trên ứng dụng Google Maps

**Mạng xã hội Facebook:** Một trong những ứng dụng phổ biến nhất của Machine Learning là đề xuất gắn thẻ bạn bè tự động trên Facebook hoặc bất kỳ nền tảng truyền thông xã hội nào khác. Facebook sử dụng tính năng nhận diện khuôn mặt và nhận dạng hình ảnh để tự động tìm thấy khuôn mặt của người phù hợp với cơ sở dữ liệu của họ và do đó đề nghị người dùng gắn thẻ người đó trên DeepFace. Dự án DeepFace của facebook thực hiện việc nhận diện khuôn mặt và xác định đối tượng cụ thể trong ảnh.




Hình 1.7: Gắn thẻ, nhận dạng của Machine Learning

## 1.2. Phân lớp nhị phân và mô hình SVM

### 1.2.1. Phân lớp nhị phân

**Khái niệm:** Phân lớp nhị phân (Binary classification) là nhiệm vụ phận loại các phần tử của một tập hợp các đối tượng ra thành 2 nhóm dựa trên cơ sở là chúng có một thuộc tính nào đó hay không (hay còn gọi là tiêu chí). Một số nhiệm vụ phân loại nhị phân điển hình:

- Kiểm tra y khoa xem một bệnh nhân có bệnh nào đó hay không (thuộc tính để phân loại là căn bệnh đó).
- Quản lý chất lượng trong nhà máy: xác định xem một sản phẩm làm ra là đủ tốt để bán chưa, hay nên loại bỏ nó (thuộc tính để phân loại là tính đủ tốt).
- Xác định xem một trang hay một bài báo có nên nằm trong tập kết quả của một truy vấn hay không (thuộc tính là độ liên quan của bài báo - thường là sự hiện diện của một số từ nào đó trong bài báo đó).

**Đánh giá bộ phân lớp nhị phân:** Để đánh giá độ hiệu quả của một xét nghiệm y khoa, người ta thường sử dụng các khái niệm độ nhạy và đặc trưng. Giả sử chúng ta xét nghiệm xem một vài người nào đó có bệnh hay không.

- Một số người có bệnh, và kết quả xét nghiệm là dương tính (positive). Họ được gọi là các dương tính đúng.
- Một số người có bệnh, nhưng kết quả xét nghiệm âm tính (negative). Họ được gọi là các âm tính sai.
- Một số không có bệnh, và kết quả xét nghiệm cũng là âm tính. Họ được gọi là các âm tính đúng.
- Một số không có bệnh, nhưng kết quả xét nghiệm lại là dương tính. Họ được gọi là các dương tính sai.
- Tổng số người dương tính đúng, âm tính đúng, dương tính sai, âm tính sai chiếm 100% tổng số người được xét nghiệm.

**Độ nhạy** (sensitivity): là tỉ lệ của số người bị bệnh được xác định đúng là có bệnh trên tổng số người bị bệnh. Nó có thể được coi là "xác suất xét nghiệm cho kết quả dương tính khi người được xét nghiệm có bị bệnh". Độ nhạy càng cao, càng ít khả năng bệnh không được phát hiện.

**Đặc trưng** (specificity): là tỉ lệ của số người không bị bệnh có kết quả xét nghiệm âm tính trên tổng số người không có bệnh (thực), nghĩa là (âm tính đúng)/ (âm tính đúng + dương tính sai). Nó còn được coi là xác suất xét nghiệm cho kết quả âm tính đối với người không có bệnh. Độ đặc trưng càng cao, càng ít người mạnh khỏe được coi là bị bệnh (hoặc trong trường hợp nhà máy, càng ít tiền bị tốn phí do loại bỏ các sản phẩm chất lượng tốt thay vì đem bán chúng).

Về mặt lý thuyết, độ nhạy và đặc trưng là độc lập, tức là cả hai đều có thể đạt đến 100%. Trong thực tế, chúng ta phải đánh đổi cái này để được cái kia - cái này tốt lên thì cái kia xấu đi, không thể đạt được cả hai.

Một điểm cần chú ý nữa, là độ nhạy và đặc trưng là độc lập với tỉ lệ giữa số cá thể âm tính và số cá thể dương tính. Tuy nhiên, giá trị của chúng thì lại phụ thuộc vào tổng số cá thể kiểm tra (population).

### 1.2.2. Mô hình SVM

#### 1.2.2.1. Giới thiệu

Support Vector Machine (SVM) là kỹ thuật mới đối với việc phân lớp dữ liệu, là phương pháp học sử dụng không gian giả thuyết các hàm tuyến tính trên không gian giả thuyết các hàm tuyến tính trên không gian đặc trưng nhiều chiều, dựa trên lý thuyết tối ưu và lý thuyết thống kê.

Support Vector Machine (SVM) là phương pháp mạnh và chính xác nhất trong số các thuật toán nổi bật ở lĩnh vực khai phá dữ liệu. SVM bao gồm support vector classifier (SVC), bộ phân lớp dựa theo hỗ trợ và support vector regressor (SRV), bộ hồi quy dựa theo vector hỗ trợ.

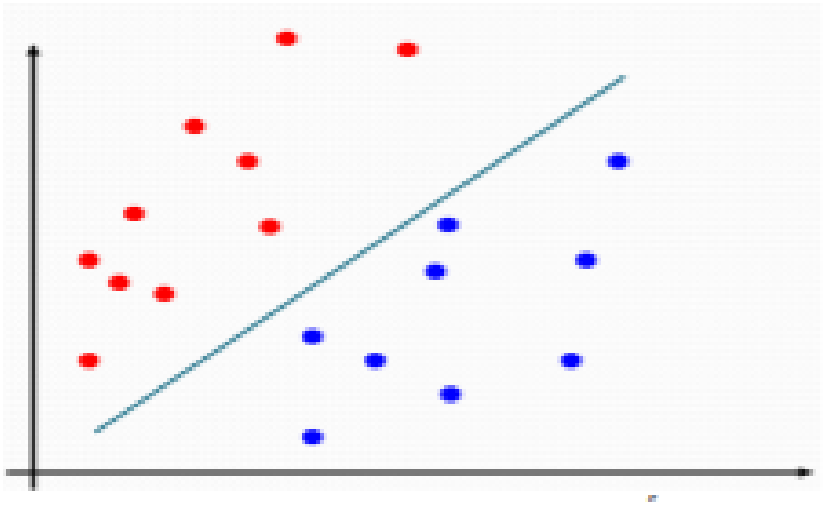
Được phát triển đầu tiên bởi Vapnik vào những năm 1990, SVM có nền tảng lý thuyết được xây dựng trên nền móng lý thuyết xác suất thống kê. Nó yêu cầu số lượng mẫu huấn luyện không nhiều và thường không nhạy cảm với số chiều của dữ liệu. Trong nhiều thập niên qua, SVM đã phát triển nhanh chóng cả về mặt lý thuyết lẫn thực nghiệm.

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

SVM Là một thuật toán học có giám sát (supervied learning) được sử dụng cho phân lớp dữ liệu, là một phương pháp thử nghiệm, đưa ra một trong những phương pháp mạnh và chính xác nhát trong số các thuật toán nổi tiếng về phân lớp dữ liệu, là một phương pháp có tính tổng quát cao nên có thể áp dụng cho nhiều loại bài toán nhận dạng và phân loại.

### 1.2.2.2. Support vector classifier phân lớp nhị phân với SVC

Xét một ví dụ của bài toán phân lớp như hình vẽ, ở đó ta phải tìm một đường thẳng sao cho phía bên trái toàn các điểm màu đỏ, bên phải nó toàn các điểm màu xanh. Bài toán mà dùng được thẳng để phân chia này được gọi là phân lớp tuyến tính (linear classification)



Hình 1.8: Minh hoạ phân tách tuyến tính

Hàm tuyến tính phân biệt bởi hai lớp như sau: $yx = w^T x + b$

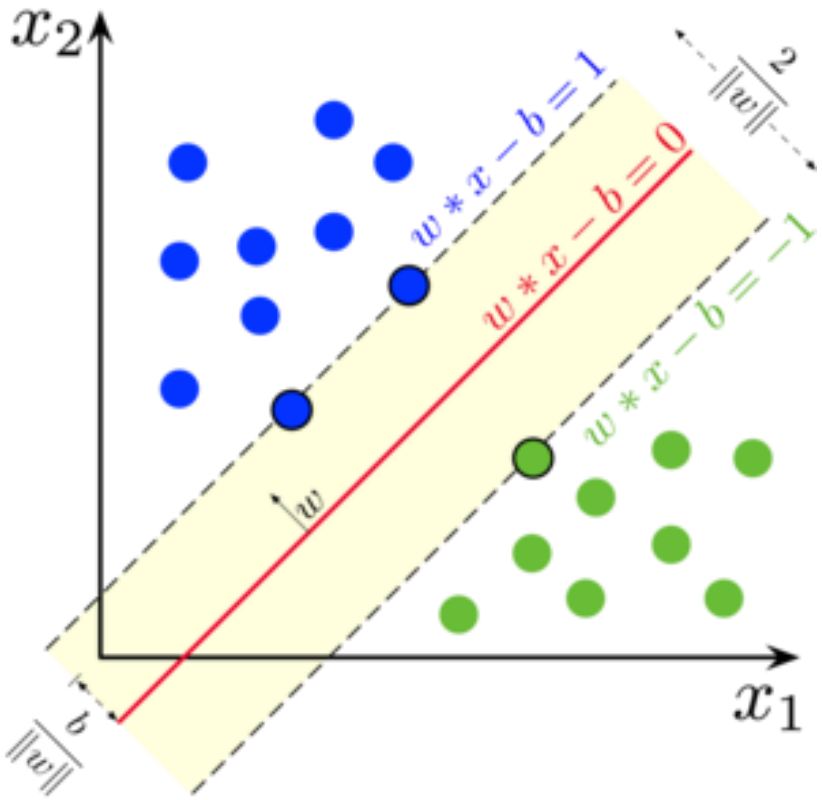
Trong đó:

- $w \in R^m$ là vector trọng số hay vector chuẩn của siêu phẳng phân cách.
- T là kí hiệu chuyển vị.
- $b \in R$ là độ lệch.

Lưu ý: với không gian hai chiều thì đường thẳng phân cách là đường thẳng, nhưng đối với không gian đa chiều thì đường thẳng phân cách gọi là siêu phẳng.

Tập dữ liệu đầu vào gồm N mẫu input vector {x1, x2, …, xn}, với các giá trị nhãn tương ứng là {t1, t2, …, tn} trong đó $t \in \{-1, 1\}$. Giả sử tập dữ liệu có thể phân tách tuyến tính hoàn toàn (nghĩa là các mẫu đều được phân tách đúng lớp bởi đường thẳng phân cách) khi đó, giá trị tham số w và b theo $yx = w^T x + b$ luôn tồn tại và thoả mãn $y(x1) > 0$ cho nhưng điểm có nhãn là $t = +1$ và $y(x1) < 0$ cho những điểm có $t = -1$, vì thế mà $t_i y(x_i) > 0$ cho mọi điểm dữ liệu huấn luyện.

Để tìm đường phân cách, SVC thông qua khái niệm gọi là lề, đường biên (margin). Lề là khoảng cách nhỏ nhất giữa điểm dữ liệu gần nhất đến một điểm bất kỳ trên đường phân cách




Hình 1.9: Margin trong SVM

Theo SVC, đường phân cách tốt nhất là đường có margin lớn nhất. điều này có nghĩa là tồn tại rất nhiều phương pháp xoay theo các phương khác nhau và khi đó phương pháp sẽ chọn ra đường phân cách mà có margin lớn nhất tức là cần tìm ra một mặt phẳng $H_0$: y = wx+b =0 và 2 siêu phẳng $H_+$, $H_-$ hỗ trợ song song với $H_0$ và có cùng khoảng cách đến $H_0$. Với điều kiện không có phần tử nào nằm giữa $H_+$, $H_-$ khi đó:

$H_+$: $w^T x+b >=1$ với t = +1

$H_-$: $w^T x+b >=-1$ với t = -1

Kết hợp 2 điều kiện trên có $y(w^T+b)>=1$ Khoảng cách từ $H_+$ và $H_-$ đến $H_0$ là $\frac{1}{||w||}$. Cần tìm siêu phẳng với lề lớn nhất, là giải bài toán tối ưu tìm cho w và b sao cho $\frac{2}{||w||}$ đạt cực đại với ràng buộc $y_i(w^T x_i+b)>=1$. Tương đương với bài toán cực tiểu hoá $\frac{w.w}{2}$ với điều kiện $y_i(w^T x_i+b)>=1$ với mọi i thuộc từ 1 đến n.

Lời giải cho bài toán tối ưu này là cực tiểu hoá hàm Lagrange:

$$L(w, b, a) = \frac{1}{2}||w.\ w|| - \sum_{i=1}^{n} ai[yi(||wTxi|| + b) - 1]$$

Trong đó, a là hệ số Lagrange, a>=0.

# CHƯƠNG 2: ỨNG DỤNG CỦA MÔ HÌNH SVM TRONG PHÂN LỚP NHỊ PHÂN

## 2.1. Mô tả bài toán

### 2.1.1. Bài toán dự đoán khả năng sống sót của bệnh nhân suy tim

#### 2.1.1.1. Mô tả

Theo thống kê, các bệnh về tim mạch (CVDs) là nguyên nhân gây ra lỗi tử vong số 1 trên toàn cầu, cướp đi sinh mạng của khoảng 17, 9 triệu người mỗi năm, chiếm 31% tổng số ca tử vong trên toàn thế giới. Những người mắc bệnh tim mạch hoặc những có nguy cơ mắc bệnh tim mạch cao (do sự hiện diện của một hoặc nhiều yếu tố nguy cơ như tăng huyết áp, tiểu đường, tăng lipid máu hoặc bệnh đã có sẵn) cần được phát hiện và quản lý sớm.

Suy tim là một sự kiện phổ biến do CVDs gây ra. Việc sử dụng mô hình học máy có thể giúp ích rất nhiều trong việc dự đoán tỉ lệ tử vong do suy tim.

Bài toán Dự đoán khả năng sống sót của bệnh nhân suy tim đưa ra tất cả các thông tin, chỉ số của người mắc bệnh suy tim, từ đó làm căn cứ chuẩn đoán khả năng tỷ vong cao hay thấp của người bệnh.

Giá trị input: Thông tin, số liệu sức khỏe của người mắc bệnh suy tim.

Giá trị output: Kết quả người mắc bệnh suy tim có tỷ lệ tử vong cao hay không.

#### 2.1.1.2. Yêu cầu bài toán

- Lấy dữ liệu về thông tin, chỉ số của người bệnh.
- Trích chọn đặc trưng từ tập dữ liệu lấy được.
- Huấn luyện tập dữ liệu.
- Chuẩn đoán khả năng sống sót của người bệnh.

### 2.1.2. Bài toán dự đoán khả năng mắc bệnh ung thư vú

#### 2.1.2.1. Mô tả

Trên thế giới, ung thư vú là loại ung thư phổ biến nhất ở phụ nữ và cao thứ hai về tỷ lệ tử vong. Tỷ lệ tử vong thay đổi nhiều từ 25 – 35/100.000 dân tại Anh, Đan Mạch, Hà Lan, Hoa Kỳ và Canada đến 2 – 5/100.000 dân ở Nhật, Mehico, Venezuela.Tỷ lệ mắc bệnh hàng năm tăng cao ở các nước Bắc Mỹ và châu Âu, trong khi ở các nước châu Á và châu Phi có xu hướng thấp hơn. Ở Hoa Kỳ trong năm 1993, đã có 182.000 ca ung thư vú mới mắc ở phụ nữ. Tỷ lệ hằng năm theo ước tính 100,2 ca mới / 100.000 dân. Từ năm 1973, tỷ lệ đã tăng trung bình hằng năm 1,8%.Theo báo cáo của Tổ chức Y tế thế giới, ung thư vú đứng hàng thứ 2 tại các nước Đông Nam Á (sau ung thư phổi) và chiếm 20% trong tổng số các ung thư ở phụ nữ.

Bài toán Chẩn đoán ung thư vú được thực hiện khi phát hiện một khối u bất thường (từ việc tự kiểm tra hoặc chụp X-quang) hoặc một đốm nhỏ canxi được nhìn thấy (trên phim chụp X-quang). Sau khi phát hiện ra một khối u đáng ngờ, bác sĩ sẽ tiến hành chẩn đoán để xác định xem nó có phải là ung thư hay không

Giá trị input: Thông tin, đặc tính của người có khả năng mắc bệnh hoặc không.

Giá trị ouput: Kết quả người được chuẩn đoán có bị mắc bệnh hay không.

#### 2.1.2.2. Yêu cầu bài toán

- Lấy dữ liệu về thông tin, chỉ số của người bệnh.
- Trích chọn đặc trưng từ tập dữ liệu lấy được.
- Huấn luyện tập dữ liệu.
- Chuẩn đoán khả năng sống sót của người bệnh.

## 2.2. Môi trường thực nghiệm

### 2.2.1. Giới thiệu về ngôn ngữ python


Hình 2.1: Ngôn ngữ python

Python là ngôn ngữ lập trình được sử dụng rất phổ biến ngày nay để phát triển nhiều loại ứng dụng phần mềm khác nhau như các chương trình chạy trên desktop, server, lập trình các ứng dụng web... Ngoài ra Python cũng là ngôn ngữ ưa thích trong ngành khoa học về dữ liệu (data science) cũng như là ngôn ngữ phổ biến để xây dựng các chương trình trí tuệ nhân tạo trong đó bao gồm machine learning.

Python là ngôn ngữ dễ học: Ngôn ngữ Python có cú pháp đơn giản, rõ ràng, sử dụng một số lượng không nhiều các từ khoá, do đó Python được đánh giá là một ngôn ngữ lập trình thân thiện với người mới học.

Python là ngôn ngữ dễ hiểu: Mã lệnh (source code hay đơn giản là code) viết bằng ngôn ngữ Python dễ đọc và dễ hiểu. Ngay cả trường hợp bạn chưa biết gì về Python bạn cũng có thể suy đoán được ý nghĩa của từng dòng lệnh trong source code.

Python có tương thích cao (highly portable): Chương trình phần mềm viết bằng ngôn ngữ Python có thể được chạy trên nhiều nền tảng hệ điều hành khác nhau bao gồm Windows, Mac OSX và Linux.

### 2.2.2. Cài đặt ngôn ngữ python và bộ thư viện cho bài toán

- **Cài đặt ngôn ngữ lập trình python**: với phiên bản python tuỳ ý trên trang chủ: https://www.python.org.
- Môi trường lập trình: Pycharm.

```
(MachineLearning) D:\PowerUniversity\MachineLearning>python
Python 3.6.5 |Anaconda, Inc.| (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
```

Hình 2.2: Cài đặt thành công python

- **Cài đặt bộ thư viện**: numpy, pandas, sklearn.

+ Để tránh xung đột với các project khác, sử dụng anaconda để tạo ra một môi trường ảo bằng lệnh: conda create -n MachineLearning python=3.6.5

+ Khi tạo xong môi trường ảo, để sử dụng các thư viện trong môi trường ảo sử dụng lệnh: activate MachineLearning

+ Tiến hành cài đặt các thư viện cần thiết cho bài toán:

```
(MachineLearning) D:\PowerUniversity\MachineLearning>pip install numpy
Collecting numpy
  Using cached numpy-1.19.5-cp36-cp36m-win_amd64.whl (13.2 MB)
Installing collected packages: numpy
Successfully installed numpy-1.19.5
```

Hình 2.3: Cài đặt thư viện numpy

```
(MachineLearning) D:\PowerUniversity\MachineLearning>pip install pandas
Collecting pandas
  Using cached pandas-1.1.5-cp36-cp36m-win_amd64.whl (8.7 MB)
Requirement already satisfied: numpy>=1.15.4 in c:\users\nam10\appdata\local\continuum\anaconda3\envs
Requirement already satisfied: pytz>=2017.2 in c:\users\nam10\appdata\local\continuum\anaconda3\envs\
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\nam10\appdata\local\continuum\anaco
Requirement already satisfied: six>=1.5 in c:\users\nam10\appdata\local\continuum\anaconda3\envs\mach
Installing collected packages: pandas
Successfully installed pandas-1.1.5
```

Hình 2.4: Cài đặt thư viện pandas

```
(MachineLearning) D:\PowerUniversity\MachineLearning>pip install sklearn
Collecting sklearn
  Using cached sklearn-0.0-py2.py3-none-any.whl
Requirement already satisfied: scikit-learn in c:\users\nam10\appdata\local\continuum\anaconda3\
Requirement already satisfied: numpy>=1.13.3 in c:\users\nam10\appdata\local\continuum\anaconda3
Requirement already satisfied: joblib>=0.11 in c:\users\nam10\appdata\local\continuum\anaconda3\
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\nam10\appdata\local\continuum\an
Requirement already satisfied: scipy>=0.19.1 in c:\users\nam10\appdata\local\continuum\anaconda3
Installing collected packages: sklearn
Successfully installed sklearn-0.0
```

Hình 2.5: Cài đặt thư viện sklearn

## 2.3. Xây dựng bộ dữ liệu

- Bài toán sử dụng bộ dữ liệu Dataset được tìm thấy trên hệ thống Data Kaggle có đường dẫn như sau: https://www.kaggle.com/andrewmvd/heart-failure-clinical-data, https://www.kaggle.com/shiva20/breast-cancer

### 2.3.1. Bộ dữ liệu cho bài toán dự đoán khả năng sống sót của bệnh nhân suy tim

- Tập dữ liệu gồm 300 bệnh nhân với các chỉ số mắc bệnh khác nhau, từ đó làm căn cứ chuẩn đoán bệnh nhân có nguy cơ tử vong cao, bênh nhân có nguy cơ tử vong thấp.Bệnh viện sẽ tổng hợp toàn bộ dữ liệu và phân loại bệnh nhân theo 2 trường hợp này.
- Đặt Y là khả năng sống sót của bệnh nhân suy tim, với Y= 0 là tử vong, ngược lại Y= 1 là sống sót.
- Bộ dữ liệu gồm 12 thuộc tính bao gồm:
  + age (tuổi)
  + anaemia (thiếu máu): anaemia có giá trị là 1 nếu như bệnh nhân thiếu máu, ngược lại không thiếu máu thì có giá trị là 0.
  + creatinin phosphokinase: mức độ của enzym CPK trong máu (mcg/L)
  + diabetes (bệnh tiểu đường): nếu bệnh nhân mắc bệnh tiểu đường thì diabetes có giá trị là 1, không mắc bệnh thì diabetes có giá trị là 0.
  + high blood pressure (phân suất tống máu)
  + ejection_fraction: phần trăm máu rời khỏi tim mỗi lần co bóp.
  + platelets (huyết áp cao).
  + serum creatinine (mức độ creatinine huyết thanh trong máu).
  + serum sodium (nồng độ natri huyết thanh trong máu).
  + sex (giới tính): nếu bệnh nhân có giới tính là nam thì sex có giá trị là 1, ngược lại nếu bệnh nhân có giới tính là nữ thì sex có giá trị là 0.
  + smoking (hút thuốc): nếu bệnh nhân hút thuốc thì smoking có giá trị là 1, còn nếu bệnh nhân không hút thuốc thì có giá trị là 0.
  + time (thời gian): thời gian theo dõi.
- Bộ dữ liệu được chia thành 2 phần:

+ 80% dữ liệu dùng để huấn luyện mô hình hay cách khác gọi là tập train.

+ 20% dữ liệu dùng làm tập dữ liệu thử nghiệm hay cách khác được gọi là tập dữ liệu test.

| age | anaemia | creatinine_phosphokinase | diabetes | ejection_fraction | high_blood_pressure | platelets | serum_creatinine | serum_sodium | sex | smoking | time | DEATH_EVENT |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 75 | 0 | 582 | 0 | 20 | 1 | 265000 | 1.9 | 130 | 1 | 0 | 4 | 1 |
| 55 | 0 | 7861 | 0 | 38 | 0 | 263358.03 | 1.1 | 136 | 1 | 0 | 6 | 1 |
| 65 | 0 | 146 | 0 | 20 | 0 | 162000 | 1.3 | 129 | 1 | 1 | 7 | 1 |
| 50 | 1 | 111 | 0 | 20 | 0 | 210000 | 1.9 | 137 | 1 | 0 | 7 | 1 |
| 65 | 1 | 160 | 1 | 20 | 0 | 327000 | 2.7 | 116 | 0 | 0 | 8 | 1 |
| 90 | 1 | 47 | 0 | 40 | 1 | 204000 | 2.1 | 132 | 1 | 1 | 8 | 1 |
| 75 | 1 | 246 | 0 | 15 | 0 | 127000 | 1.2 | 137 | 1 | 0 | 10 | 1 |
| 60 | 1 | 315 | 1 | 60 | 0 | 454000 | 1.1 | 131 | 1 | 1 | 10 | 1 |
| 65 | 0 | 157 | 0 | 65 | 0 | 263358.03 | 1.5 | 138 | 0 | 0 | 10 | 1 |
| 80 | 1 | 123 | 0 | 35 | 1 | 388000 | 9.4 | 133 | 1 | 1 | 10 | 1 |
| 75 | 1 | 81 | 0 | 38 | 1 | 368000 | 4 | 131 | 1 | 1 | 10 | 1 |
| 62 | 0 | 231 | 0 | 25 | 1 | 253000 | 0.9 | 140 | 1 | 1 | 10 | 1 |
| 45 | 1 | 981 | 0 | 30 | 0 | 136000 | 1.1 | 137 | 1 | 0 | 11 | 1 |
| 50 | 1 | 168 | 0 | 38 | 1 | 276000 | 1.1 | 137 | 1 | 0 | 11 | 1 |
| 49 | 1 | 80 | 0 | 30 | 1 | 427000 | 1 | 138 | 0 | 0 | 12 | 0 |
| 82 | 1 | 379 | 0 | 50 | 0 | 47000 | 1.3 | 136 | 1 | 0 | 13 | 1 |
| 87 | 1 | 149 | 0 | 38 | 0 | 262000 | 0.9 | 140 | 1 | 0 | 14 | 1 |
| 45 | 0 | 582 | 0 | 14 | 0 | 166000 | 0.8 | 127 | 1 | 0 | 14 | 1 |
| 70 | 1 | 125 | 0 | 25 | 1 | 237000 | 1 | 140 | 0 | 0 | 15 | 1 |
| 48 | 1 | 582 | 1 | 55 | 0 | 87000 | 1.9 | 121 | 0 | 0 | 15 | 1 |
| 65 | 1 | 52 | 0 | 25 | 1 | 276000 | 1.3 | 137 | 0 | 0 | 16 | 0 |
| 65 | 1 | 128 | 1 | 30 | 1 | 297000 | 1.6 | 136 | 0 | 0 | 20 | 1 |
| 68 | 1 | 220 | 0 | 35 | 1 | 289000 | 0.9 | 140 | 1 | 1 | 20 | 1 |
| 53 | 0 | 63 | 1 | 60 | 0 | 368000 | 0.8 | 135 | 1 | 0 | 22 | 0 |
| 75 | 0 | 582 | 1 | 30 | 1 | 263358.03 | 1.83 | 134 | 0 | 0 | 23 | 1 |
| 80 | 0 | 148 | 1 | 38 | 0 | 149000 | 1.9 | 144 | 1 | 1 | 23 | 1 |
| 95 | 1 | 112 | 0 | 40 | 1 | 196000 | 1 | 138 | 0 | 0 | 24 | 1 |
| 70 | 0 | 122 | 1 | 45 | 1 | 284000 | 1.3 | 136 | 1 | 1 | 26 | 1 |
| 58 | 1 | 60 | 0 | 38 | 0 | 153000 | 5.8 | 134 | 1 | 0 | 26 | 1 |

Hình 2.6: Bộ dữ liệu dự đoán khả năng sống sót của bệnh nhân suy tim

### 2.3.2. Bộ dữ liệu cho bài toán dự đoán khả năng mắc bệnh ung thư vú

- Tập dữ liệu gồm thông tin của 570 bệnh nhân với các chỉ số khối u bất thường khác nhau, từ đó làm căc cứ chuẩn đoán bệnh nhân có bị ún thú vú hay không.
- Đặt Y là khả năng mắc bệnh ung thư vú, với Y = 0 là không mắc bệnh ung thư vú và với Y = 1 thì bệnh nhân đó bị mắc bệnh ung thư vú.
- Bộ dữ liệu gồm 5 thuộc tính
  + mean_radius (bán kính trung bình)
  + mean_texture (kết cấu trung vị)
  + mean_perimeter (chu vi trung bình)
  + mean_area (diện tích trung bình)
  + mean_smoothness (sai số trung bình).
- Bộ dữ liệu được chia thành 2 phần: 80% dữ liệu dùng để huấn luyện mô hình hay cách khác gọi là tập train, 20% dữ liệu dùng làm tập dữ liệu thử nghiệm hay cách khác được gọi là tập dữ liệu test.

| mean_radius | mean_texture | mean_perimeter | mean_area | mean_smoothness | diagnosis |
|---|---|---|---|---|---|
| 17.99 | 10.38 | 122.8 | 1001.0 | 1.184 | 0 |
| 20.57 | 17.77 | 132.9 | 1326.0 | 8.474 | 0 |
| 19.69 | 21.25 | 130.0 | 1203.0 | 1.096 | 0 |
| 11.42 | 20.38 | 77.58 | 386.1 | 1.425 | 0 |
| 20.29 | 14.34 | 135.1 | 1297.0 | 1.003 | 0 |
| 12.45 | 15.7 | 82.57 | 477.1 | 1.278 | 0 |
| 18.25 | 19.98 | 119.6 | 1040.0 | 9.463 | 0 |
| 13.71 | 20.83 | 90.2 | 577.9 | 1.189 | 0 |
| 13.0 | 21.82 | 87.5 | 519.8 | 1.273 | 0 |
| 12.46 | 24.04 | 83.97 | 475.9 | 1.186 | 0 |
| 16.02 | 23.24 | 102.7 | 797.8 | 8.206 | 0 |
| 15.78 | 17.89 | 103.6 | 781.0 | 971 | 0 |
| 19.17 | 24.8 | 132.4 | 1123.0 | 974 | 0 |
| 15.85 | 23.95 | 103.7 | 782.7 | 8.401 | 0 |
| 13.73 | 22.61 | 93.6 | 578.3 | 1.131 | 0 |
| 14.54 | 27.54 | 96.73 | 658.8 | 1.139 | 0 |
| 14.68 | 20.13 | 94.74 | 684.5 | 9.867 | 0 |
| 16.13 | 20.68 | 108.1 | 798.8 | 117 | 0 |
| 19.81 | 22.15 | 130.0 | 1260.0 | 9.831 | 0 |

Hình 2.7: Bộ dữ liệu dự đoán khả năng mắc bệnh ung thư vú

## 2.4. Áp dụng thuật toán SVM vào bài toán dự đoán khả năng sống sót của bệnh nhân suy tim

### 2.4.1. Cài đặt thuật toán

- Khai báo các thư viện cần thiết cho bài toán

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.preprocessing import StandardScaler
```

Thực hiện các chức năng sau trong hàm HeartFailure_and_BreastCancer (file_csv, classification_properties, properties).

- Đọc file dữ liệu, phân tích dữ liêu

```
df = pd.read_csv(file_csv)
print("Hiển thị 5 mẫu dữ liệu của file: \n", df.head())
X = df.drop([classification_properties], axis=1)
y = df[classification_properties]
print("Số lượng nhãn của các lớp:\n", y.value_counts())
print("Dữ liệu X: \n", X)
print("Nhãn Y: \n", y)
```

- Chuẩn hoá dữ liệu

```
'''Chuẩn hoá'''
std = StandardScaler()
X = std.fit_transform(X)
print("X sau khi được chuẩn hoá:\n", X)
```

- Vẽ dữ liệu lên không gian 3 chiều

```
'''vẽ dữ liệu sử dụng PCA lên không gian 3 chiều'''
print("Dữ liệu trước khi sử dụng PCA: ", X.shape)
X = PCA(3).fit_transform(X)
x_pca = X[:, 0]
y_pca = X[:, 1]
z_pca = X[:, 2]
fig = plt.figure()
ax = fig.add_subplot(projection='3d')
ax.scatter(x_pca, y_pca, z_pca, c=y, s=60)
ax.legend(['Malign'])
ax.set_xlabel('First Principal Component')
ax.set_ylabel('Second Principal Component')
ax.set_zlabel('Third Principal Component')
plt.show()
print("Dữ liệu sau kho giảm chiều: ", X.shape)
```

- Mối quan hệ của các thành phần chính với nhau

```
'''Biểu diễn mối quan hệ giữa các thành phần chính'''
sns.scatterplot(x=x_pca, y=z_pca, hue=y, palette='Set1')
plt.xlabel('First Principal Component')
plt.ylabel('Third Principal Component')
plt.show()

sns.scatterplot(x=y_pca, y=z_pca, hue=y, palette='Set1')
plt.xlabel('Second Principal Component')
plt.ylabel('Third Principal Component')
plt.show()

sns.scatterplot(x=x_pca, y=y_pca, hue=y, palette='Set1')
plt.xlabel('First Principal Component')
plt.ylabel('Second Principal Component')
plt.show()
```

- Chia tập train test theo tỉ lệ 8:2

```
'''Chia tập dữ liệu theo tỉ lệ 8:2'''
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=35)
print("Số dữ liệu train", len(X_train))
print("Số dữ liệu test", len(X_test))
print("Dữ liệu để train:\n", X_train)
print("Nhãn dùng để train:\n", y_train)
print("Dữ liệu dùng để test:\n", X_test)
print("Hiển thị nhãn để test:\n", y_test)
```

- Chạy mô hình học máy

```
'''Chạy mô hình học máy: huấn luyện mô hình'''
models = SVC(kernel='linear').fit(X_train, y_train)
'''Dự đoán mô hình'''
y_predict = models.predict(X_test)
print("Hệ số w", models.coef_)
print(models.coef_.shape)
print("Hệ số bias", models.intercept_)
print("Số lớp", models.classes_)
```

- Đánh giá mô hình học máy và biểu diễn mối liên hệ

```
'''Đánh giá mô hình học dựa trên kết quả dự đoán (với độ đo đơn giản Accuracy, Precision, Recall)'''
confusion_matrix1 = confusion_matrix(y_test, y_predict)
print("Ma trận dự doán:\n", confusion_matrix1)
print("Accuracy Score: ", accuracy_score(y_test, y_predict))
print(classification_report(y_test, y_predict))

X = df[[properties]]
y = df[classification_properties]
x0 = X[y == 0]
x1 = X[y == 1]
plt.plot(x0[properties], 'r^', markersize=4, alpha=.8)
plt.plot(x1[properties], 'b^', markersize=4, alpha=.8)
plt.xlabel(classification_properties)
plt.ylabel(properties)
plt.plot()
plt.show()
```

### 2.4.2. Kết quả thực nghiệm

- In 5 mẫu đầu tiên của tập dữ liệu và số lượng nhãn của các lớp

```
Hiển thị 5 mẫu dữ liệu của file:
    age  anaemia  creatinine_phosphokinase  ...  smoking  time  DEATH_EVENT
0  75.0        0                       582  ...        0     4            1
1  55.0        0                      7861  ...        0     6            1
2  65.0        0                       146  ...        1     7            1
3  50.0        1                       111  ...        0     7            1
4  65.0        1                       160  ...        0     8            1

[5 rows x 13 columns]
Số lượng nhãn của các lớp:
 0    203
1     96
```

- Dữ liệu đầu vào X:

```
Dữ liệu X:
      age  anaemia  creatinine_phosphokinase  ...  sex  smoking  time
0    75.0        0                       582  ...    1        0     4
1    55.0        0                      7861  ...    1        0     6
2    65.0        0                       146  ...    1        1     7
3    50.0        1                       111  ...    1        0     7
4    65.0        1                       160  ...    0        0     8
..    ...      ...                       ...  ...  ...      ...   ...
294  62.0        0                        61  ...    1        1   270
295  55.0        0                      1820  ...    0        0   271
296  45.0        0                      2060  ...    0        0   278
297  45.0        0                      2413  ...    1        1   280
298  50.0        0                       196  ...    1        1   285
```

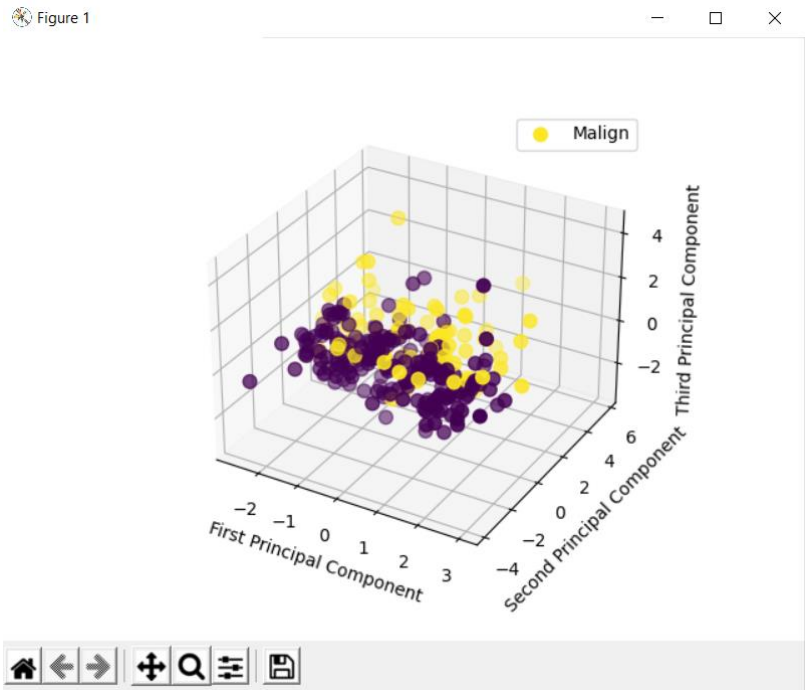
- Dữ liệu nhãn đầu ra Y:

```
Dữ liệu Y:
 0      1
1      1
2      1
3      1
4      1
      ..
294    0
295    0
296    0
297    0
298    0
Name: DEATH_EVENT, Length: 299, dtype: int64
```

- Dữ liệu X sau khi chuẩn hoá

```
X sau khi được chuẩn hoá:
 [[ 1.19294523e+00 -8.71104775e-01  1.65728387e-04 ...  7.35688190e-01
  -6.87681906e-01 -1.62950241e+00]
 [-4.91279276e-01 -8.71104775e-01  7.51463953e+00 ...  7.35688190e-01
  -6.87681906e-01 -1.60369074e+00]
 [ 3.50832977e-01 -8.71104775e-01 -4.49938761e-01 ...  7.35688190e-01
   1.45416070e+00 -1.59078490e+00]
 ...
 [-1.33339153e+00 -8.71104775e-01  1.52597865e+00 ... -1.35927151e+00
  -6.87681906e-01  1.90669738e+00]
 [-1.33339153e+00 -8.71104775e-01  1.89039811e+00 ...  7.35688190e-01
   1.45416070e+00  1.93250906e+00]
 [-9.12335403e-01 -8.71104775e-01 -3.98321274e-01 ...  7.35688190e-01
   1.45416070e+00  1.99703825e+00]]
```

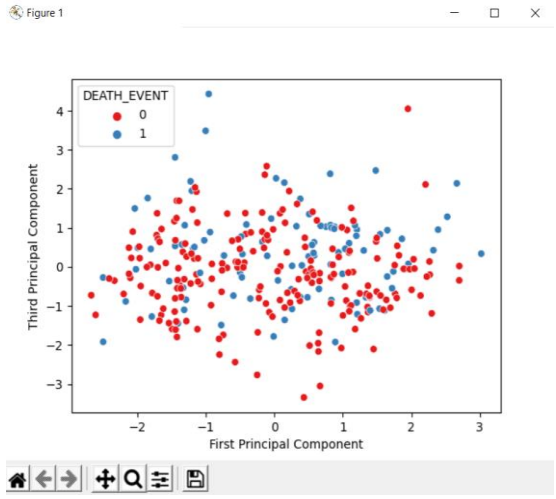
- Biểu diễn dữ liệu trên không gian 3D

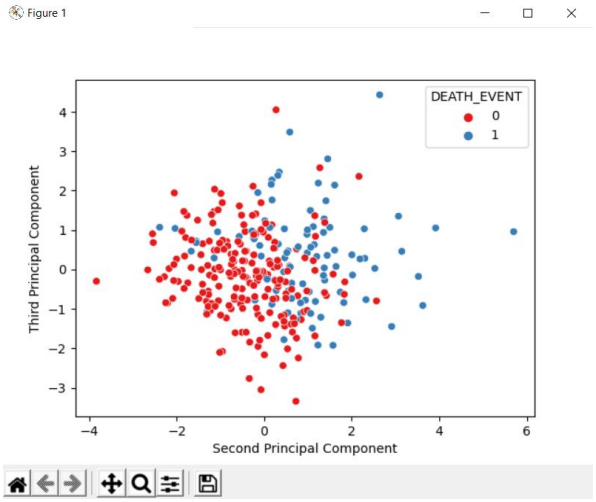


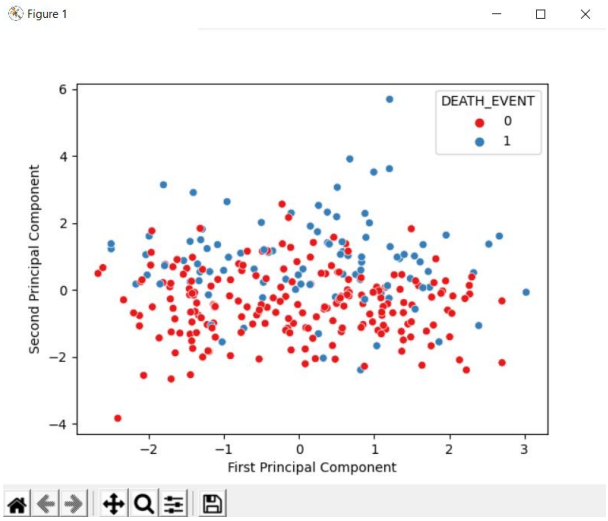
- Số chiều sau khi sử dụng phương pháp PCA để giảm chiều dữ liệu

```
Dữ liệu trước khi sử dụng PCA:  (299, 12)
Dữ liệu sau kho giảm chiều:  (299, 3)
Số dữ liệu train 239
Số dữ liệu test 60
```

- Mối quan hệ giữa các thành phần chính




- Dữ liệu X_train:

```
Dữ liệu để train:
 [[ 8.77054673e-01  2.28710374e+00  1.03322965e+00]
 [-6.18837903e-01 -1.01847482e+00  6.69715445e-01]
 [-1.22474337e+00 -1.03735062e+00 -4.23014724e-01]
 [-1.65503674e+00 -8.67112653e-01 -1.22964500e+00]
 [-2.12647417e+00 -7.66061912e-01 -1.76978577e-01]
 [-1.52386021e-01 -1.14120294e+00  4.88736147e-01]
 [ 3.39176036e-01 -5.09440120e-01 -7.22652955e-01]
 [-4.38884344e-01  3.38442672e-01  1.10889179e-01]
 [ 1.36770409e+00 -1.83304557e+00 -4.81942048e-01]
 [ 1.78896151e+00 -1.04887190e+00 -7.92802559e-01]
 [ 2.39225718e+00 -1.06583583e+00  9.54596134e-01]
```

- Dữ liệu Y_train

```
Nhãn dùng để train:
 0      1
285    0
92     0
64     0
240    0
      ..
232    0
249    0
33     0
271    0
201    0
Name: DEATH_EVENT, Length: 239, dtype: int64
```

- Dữ liệu dùng để thử nghiệm:

```
Dữ liệu dùng để test:
 [[ 0.51437434  0.53190983 -2.0130545 ]
 [-0.47543144  0.22979344 -0.26435802]
 [ 1.48671874 -1.40747576  0.65290667]
 [-0.91500145 -1.96612625 -0.3029075 ]
 [-0.01174932  0.4501421  -1.78007494]
 [ 1.82059807  0.93013522  0.29760518]
 [-0.80990687 -0.3325036  -1.84284535]
 [ 1.54471874 -0.57272084  0.83618494]
 [ 1.4809515   0.34245731  2.47094538]
 [-0.17961334 -1.2067536   1.39334508]
 [ 1.33807772  0.94082206 -0.7790636 ]
 [ 1.38793937 -0.67950469 -0.75699343]
 [-1.13992549 -0.99109892  1.92824741]
 [-1.43158073 -1.52335325  1.25102018]
 [ 0.57956572  1.14323214  0.63117444]
 [ 1.03298015  0.08603273 -0.50138105]
 [ 0.0811968  -2.20933653 -0.84890319]
 [-1.12791358 -0.46620209  1.13446544]
 [ 0.88987408  1.5718772  -1.92315687]
 [-0.09281885  0.23990976  0.69273589]
 [-2.09213191  0.30297045  0.23078466]
 [ 0.98517855 -0.06955105  1.01481139]
 [-1.20750173 -0.14617481  1.95043436]
 [-1.02614204 -1.55525891  0.68153166]
 [-1.70419587 -2.66262391 -0.0096911 ]
 [ 1.50250612 -0.45058145  0.2182599 ]
```

- Hiển thị 10 nhãn Y_test:

```
Hiển thị 10 nhãn để test:
 159    0
68     1
153    0
250    0
195    1
203    0
254    0
182    1
17     1
258    0
```

- Dự đoán mô hình học máy:

```
Kết quả dự đoán:
 [0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0
 1 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0]
Hệ số w:  [[0.09712721 0.9197461  0.39186704]]
(1, 3)
Hệ số bias:  [-0.87668272]
Số lớp:  [0 1]
```

- Đánh giá kết quả

- Sau khi thử nghiệm với bộ dữ liệu thì kết quả phân lớp đạt 71,67%.

```
Accuracy Score:  0.7166666666666667
```

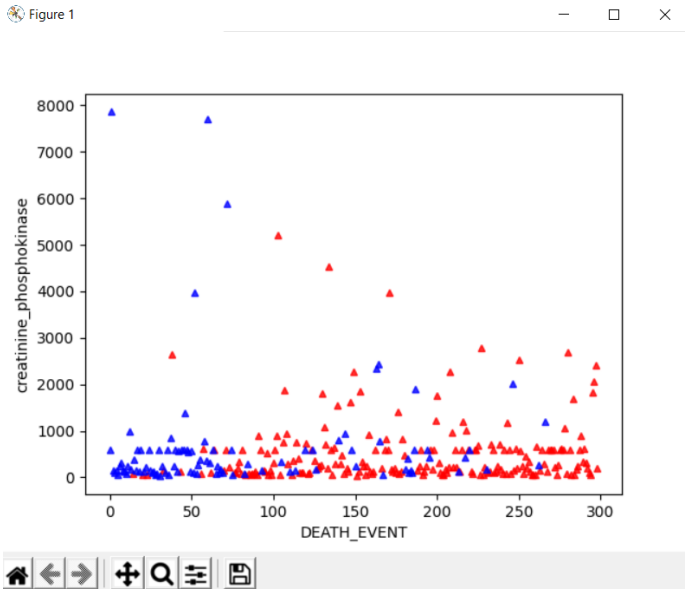
- Đối với Precision, Recall:

```
              precision    recall  f1-score   support

           0       0.74      0.90      0.81        41
           1       0.60      0.32      0.41        19

    accuracy                           0.72        60
   macro avg       0.67      0.61      0.61        60
weighted avg       0.70      0.72      0.69        60
```

- Ma trận dự đoán:

```
Ma trận dự doán:
 [[37  4]
 [13  6]]
```

- Sử dụng matplotlib để vẽ:




## 2.5. Áp dụng thuật toán SVM vào bài toán dự đoán khả năng mắc bệnh ung thư vú

### 2.5.1. Cài đặt thuật toán

- Khai báo các thư viện cần thiết cho bài toán

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.preprocessing import StandardScaler
```

- Thực hiện các chức năng sau trong hàm HeartFailure_and_BreastCancer (file_csv, classification_properties, properties).

- Đọc file dữ liệu, phân tích dữ liêu

```
df = pd.read_csv(file_csv)
print("Hiển thị 5 mẫu dữ liệu của file: \n", df.head())
X = df.drop([classification_properties], axis=1)
y = df[classification_properties]
print("Số lượng nhãn của các lớp:\n", y.value_counts())
print("Dữ liệu X: \n", X)
print("Nhãn Y: \n", y)
```

- Chuẩn hoá dữ liệu

```
'''Chuẩn hoá'''
std = StandardScaler()
X = std.fit_transform(X)
print("X sau khi được chuẩn hoá:\n", X)
```

- Vẽ dữ liệu lên không gian 3 chiều

```
'''vẽ dữ liệu sử dụng PCA lên không gian 3 chiều'''
print("Dữ liệu trước khi sử dụng PCA: ", X.shape)
X = PCA(3).fit_transform(X)
x_pca = X[:, 0]
y_pca = X[:, 1]
z_pca = X[:, 2]
fig = plt.figure()
ax = fig.add_subplot(projection='3d')
ax.scatter(x_pca, y_pca, z_pca, c=y, s=60)
ax.legend(['Malign'])
ax.set_xlabel('First Principal Component')
ax.set_ylabel('Second Principal Component')
ax.set_zlabel('Third Principal Component')
plt.show()
print("Dữ liệu sau kho giảm chiều: ", X.shape)
```

- Mối quan hệ của các thành phần chính với nhau

```
'''Biểu diễn mối quan hệ giữa các thành phần chính'''
sns.scatterplot(x=x_pca, y=z_pca, hue=y, palette='Set1')
plt.xlabel('First Principal Component')
plt.ylabel('Third Principal Component')
plt.show()

sns.scatterplot(x=y_pca, y=z_pca, hue=y, palette='Set1')
plt.xlabel('Second Principal Component')
plt.ylabel('Third Principal Component')
plt.show()

sns.scatterplot(x=x_pca, y=y_pca, hue=y, palette='Set1')
plt.xlabel('First Principal Component')
plt.ylabel('Second Principal Component')
plt.show()
```

- Chia tập train test theo tỉ lệ 8:2

```
'''Chia tập dữ liệu theo tỉ lệ 8:2'''
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=35)
print("Số dữ liệu train", len(X_train))
print("Số dữ liệu test", len(X_test))
print("Dữ liệu để train:\n", X_train)
print("Nhãn dùng để train:\n", y_train)
print("Dữ liệu dùng để test:\n", X_test)
print("Hiển thị nhãn để test:\n", y_test)
```

- Chạy mô hình học máy

```
'''Chạy mô hình học máy: huấn luyện mô hình'''
models = SVC(kernel='linear').fit(X_train, y_train)
'''Dự đoán mô hình'''
y_predict = models.predict(X_test)
print("Hệ số w", models.coef_)
print(models.coef_.shape)
print("Hệ số bias", models.intercept_)
print("Số lớp", models.classes_)
```

- Đánh giá mô hình học máy

```
'''Đánh giá mô hình học dựa trên kết quả dự đoán (với độ đo đơn giản Accuracy, Precision, Recall)'''
print("Accuracy Score: \n", accuracy_score(y_test, y_predict))
print(classification_report(y_test, y_predict))
confusion_matrix1 = confusion_matrix(y_test, y_predict)
print("Ma trận dự đoán: \n", confusion_matrix1)

X = df[[properties]]
y = df[classification_properties]
x0 = X[y == 0]
x1 = X[y == 1]
plt.plot(x0[properties], 'b^', markersize=4, alpha=.8)
plt.plot(x1[properties], 'go', markersize=4, alpha=.8)
plt.xlabel('')
plt.ylabel(properties)
plt.plot()
plt.show()
```

### 2.5.2. Kết quả thực nghiệm

- In 5 mẫu đầu tiên của tập dữ liệu và số lượng nhãn của các lớp

```
Hiển thị 5 mẫu dữ liệu của file:
    mean_radius  mean_texture  ...  mean_smoothness  diagnosis
0         17.99         10.38  ...          0.11840          0
1         20.57         17.77  ...          0.08474          0
2         19.69         21.25  ...          0.10960          0
3         11.42         20.38  ...          0.14250          0
4         20.29         14.34  ...          0.10030          0

[5 rows x 6 columns]
Số lượng nhãn của các lớp:
 1    357
0    212
```

- Dữ liệu đầu vào X:

```
Dữ liệu X:
      mean_radius  mean_texture  mean_perimeter  mean_area  mean_smoothness
0           17.99         10.38          122.80     1001.0          0.11840
1           20.57         17.77          132.90     1326.0          0.08474
2           19.69         21.25          130.00     1203.0          0.10960
3           11.42         20.38           77.58      386.1          0.14250
4           20.29         14.34          135.10     1297.0          0.10030
..            ...           ...             ...        ...              ...
564         21.56         22.39          142.00     1479.0          0.11100
565         20.13         28.25          131.20     1261.0          0.09780
566         16.60         28.08          108.30      858.1          0.08455
567         20.60         29.33          140.10     1265.0          0.11780
568          7.76         24.54           47.92      181.0          0.05263
```

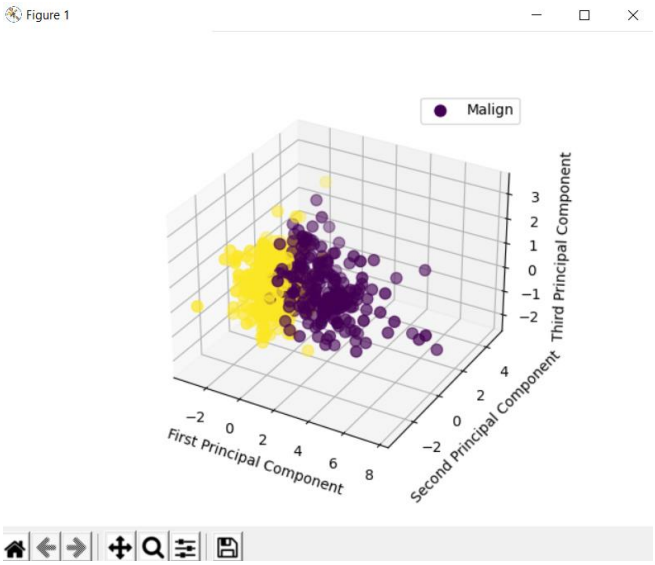
- Dữ liệu nhãn đầu ra Y:

```
Nhãn Y:
 0      0
1       0
2       0
3       0
4       0
       ..
564     0
565     0
566     0
567     0
568     1
```

- Dữ liệu X sau khi chuẩn hoá

```
X sau khi được chuẩn hoá:
 [[ 1.09706398 -2.07333501  1.26993369  0.9843749   1.56846633]
 [ 1.82982061 -0.35363241  1.68595471  1.90870825 -0.82696245]
 [ 1.57988811  0.45618695  1.56650313  1.55888363  0.94221044]
 ...
 [ 0.70228425  2.0455738   0.67267578  0.57795264 -0.84048388]
 [ 1.83834103  2.33645719  1.98252415  1.73521799  1.52576706]
 [-1.80840125  1.22179204 -1.81438851 -1.34778924 -3.11208479]]
```

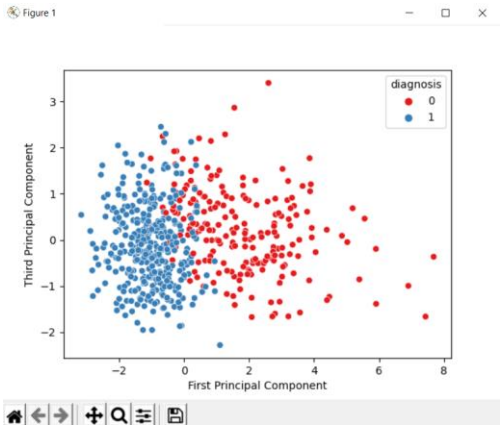
- Biểu diễn dữ liệu trên không gian 3D

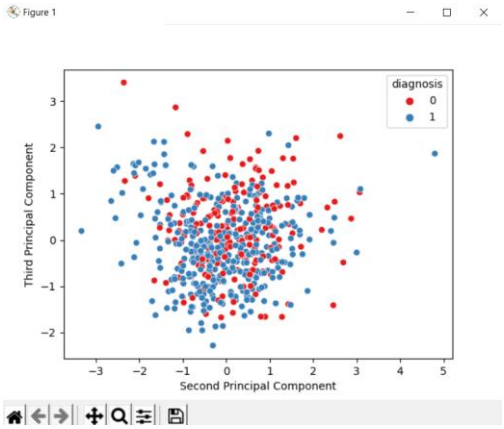


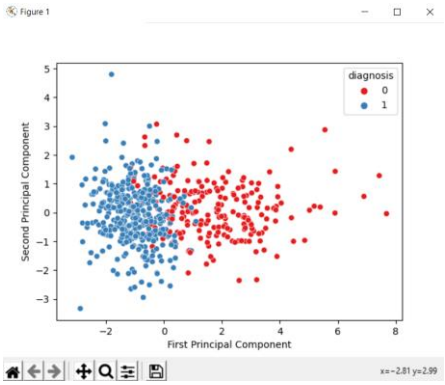
- Số chiều sau khi sử dụng phương pháp PCA để giảm chiều dữ liệu

```
Dữ liệu trước khi sử dụng PCA:  (569, 5)
Dữ liệu sau kho giảm chiều:  (569, 3)
Số dữ liệu train 455
Số dữ liệu test 114
```

- Mối quan hệ giữa các thành phần chính




- Dữ liệu X_train:

```
[[-1.30619009e+00  1.63358210e+00  8.51419641e-01]
[-9.75915714e-01 -8.20325120e-01  1.09173138e+00]
[-7.55730339e-01  6.45381547e-01 -2.84236984e-01]
...
[ 2.69527217e-03 -7.55024100e-01 -4.83766042e-01]
[ 2.81620450e+00 -9.94695459e-01  5.50312624e-01]
[-1.65627851e+00  8.74213236e-01 -7.16055057e-01]]
```

- Dữ liệu Y_train

```
Nhãn dùng để train:
 507    1
559    1
292    1
396    1
312    1
      ..
184    0
249    1
448    1
33     0
271    1
```

- Dữ liệu dùng để thử nghiệm:

```
Dữ liệu dùng để test:
 [[-1.83447451e+00  2.51553361e-01  9.11669854e-01]
 [-5.65298017e-01  9.82556948e-01  2.30212214e+00]
 [ 2.63065744e-01  1.48300262e+00 -1.40542004e+00]
 [-2.43759103e+00  2.99161770e-01 -7.47646123e-02]
 [-1.23325348e+00  9.98854849e-01  4.65386666e-01]
 [ 2.85765277e-01 -2.20826180e-01 -2.95698714e-01]
 [-1.11753084e+00 -6.32481973e-02  2.93421994e-02]
 [-2.34778961e+00  7.10677941e-01  9.84994601e-01]
 [ 3.20990637e-01  5.70957692e-01 -1.26266143e+00]
 [-7.11781383e-01 -2.94182799e+00  2.45337333e+00]
 [-1.48985482e-01  1.09673977e-01 -1.19061615e+00]
 [-6.75228519e-01  1.28218918e+00  3.06872486e-01]
 [ 2.36214433e+00 -9.91835187e-01  1.49395758e-01]
 [ 3.92972630e+00  9.08342400e-01  2.59951532e-01]
 [-1.22225566e+00  3.60043746e-01 -1.43760390e+00]
 [-1.82219557e+00  2.06142126e-01  7.23191871e-01]
 [ 3.98121074e-01 -1.57752796e+00  6.01856425e-01]
 [-3.16177906e+00  1.92005056e+00  5.42788202e-01]
 [-8.48505137e-01 -1.25906977e+00 -8.58357400e-01]
 [ 2.19995778e+00 -8.41442126e-01  1.28845794e+00]
 [ 2.57306809e-01 -1.11648734e+00  1.61640603e-01]
 [ 3.22242871e+00  1.78275602e-01  4.49329281e-01]
 [-9.89994429e-01  5.73435768e-01  2.28809779e-01]
 [-6.13413908e-01 -1.42769543e+00  2.12216487e+00]
```

- Hiển thị 10 nhãn Y_test:

```
Hiển thị nhãn để test:
 97     1
537    1
484    1
116    1
142    1
      ..
92     1
539    1
565    0
213    0
481    1
```

- Dự đoán mô hình học máy:

```
Kết quả dự đoán:
 [1 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0 1 1 0 1 0 1 1 0 1 0 0 1 1 1 0 0 0 1 1 1
 1 1 1 1 1 1 0 1 1 0 0 0 0 1 1 1 0 0 0 1 1 1 1 0 0 0 1 0 0 0 1 0 1 1 1 0 1
 0 1 1 1 0 1 1 1 1 1 1 0 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 0 1 0 0 1 1 1 1 1 1
 0 0 1]
Hệ số w:  [[-2.15544602 -0.4256363  -0.95025021]]
(1, 3)
Hệ số bias:  [0.64165227]
Số lớp:  [0 1]
```

- Đánh giá kết quả

- Sau khi thử nghiệm với bộ dữ liệu thì kết quả phân lớp đạt 93,85%.

```
Accuracy Score:  0.9385964912280702
```

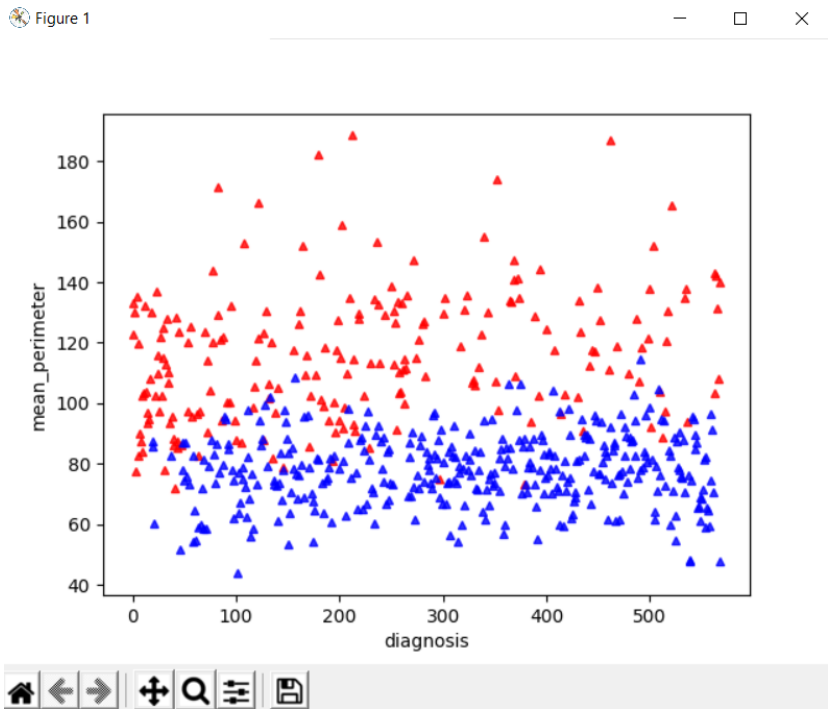
- Đối với Precision, Recall:

```
              precision    recall  f1-score   support

           0       0.95      0.88      0.91        41
           1       0.93      0.97      0.95        73

    accuracy                           0.94       114
   macro avg       0.94      0.93      0.93       114
weighted avg       0.94      0.94      0.94       114
```

- Ma trận dự đoán:

```
Ma trận dự doán:
 [[36  5]
 [ 2 71]]
```

- Sử dụng matplotlib để vẽ:



# KẾT LUẬN

Qua việc thực hiện nghiên cứu đề tài "**Bài toán phân lớp nhị phân sử dụng SVM để dự đoán khả năng sống sót của bệnh nhân suy tim, khả năng mắc bệnh ung thư vú**", em đã được biết thêm rất nhiều kiến thức về thuật toán.

Trong quá trình thực hiện đề tài có rất nhiều ý tưởng hay và độc đáo. Nhưng do kiến thức của em còn hạn hẹp và thời gian không cho phép nên em chưa thể thực hiện được những ý tưởng đó. Tuy nhiên em đã cố gắng để xây dựng một chương trình hoàn chỉnh và đẹp nhất để đưa tới thầy, cô. Trong quá trình xây dựng chương trình em khó tránh khỏi những sai xót còn tồn tại. Vì vậy em rất mong được nhận lời góp ý và chỉnh sửa từ thầy, cô để có thể hoàn thành chương trình một cách hoàn chỉnh nhất.

Em một lần nữa xin cảm ơn giảng viên Ngô Hoàng Huy đã tận tình giảng dạy cũng như hướng dẫn chúng em làm sản phẩm kết thúc học phần trong môn học Nhập môn học máy, thầy đã giúp đỡ chúng em trong quá trình nghiên cứu đề tài và chia sẻ những tài liệu hay cũng như các kỹ năng lập trình cần thiết.

# TÀI LIỆU THAM KHẢO

[1]. Giáo trình Machine Learning cơ bản-Vũ Hữu Tập, Nhà xuất bản khoa học và kỹ thuật.

[2]. Slide bài giảng Học Máy- Nguyễn Nhật Quang, Trường Đại Học Bách Khoa Hà Nội.

[3]. Trang web Kaggle.com.

[4]. Vũ Hữu Tiệp; Machine Learning cơ bản; Last update: March8, 2018.

[5]. Nguyễn Thanh Tuấn; Deep Learning cơ bản; Last update: October 2019.

[6] https://en.wikipedia.org/wiki/Support-vector_machine.