TRƯỜNG ĐẠI HỌC ĐIỆN LỰC

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO CHUYÊN ĐỀ HỌC PHẦN
HỌC MÁY NÂNG CAO

ĐỀ TÀI: TRỰC QUAN HÓA PCA - ỨNG DỤNG PCA TRONG BỘ DỮ LIỆU DIGIT VÀ DỰ ĐOÁN KHẢ NĂNG MẮC BỆNH UNG THƯ VÚ

Sinh viên thực hiện	**: NGUYỄN VĂN NAM**
	HÀ QUÝ ĐỨC
	VŨ QUANG HUY
Giảng viên hướng dẫn	**: PHẠM ĐỨC HỒNG**
Ngành	**: CÔNG NGHỆ THÔNG TIN**
Chuyên ngành	**: CÔNG NGHỆ PHẦN MỀM**
Lớp	**: D13CNPM5**
Khóa	**: 2018 - 2023**

Hà Nội, tháng 10 năm 2020

PHIẾU CHẤM ĐIỂM

Sinh viên thực hiện:

Họ và tên sinh viên	Nội dung thực hiện	Chữ ký	Điểm
Nguyễn Văn Nam (Nhóm trưởng) 18810310428	+ Chương 1. + Chương 2. + Chương 3. + support nhóm hoàn thành báo, code.	Nam	
Hà Qúy Đức 18810310435	+ Chương 1. + Chương 2. + Chương 3.	Đức	
Vũ Quang Huy 18810310453	+ Chương 1. + Chương 2. + Chương 3.	Huy	

Họ và tên giảng viên	Chữ ký	Ghi chú
Giảng viên chấm 1:		
Giảng viên chấm 2:		

MỤC LỤC

Contents

LỜI CẢM ƠN

Nhóm chúng em xin chân thành cảm ơn các thầy, cô giáo trong Khoa Công nghệ thông tin, trường Đại học Điện Lực, đã tạo điều kiện cho em thực hiện đề tài này.

Để có thể hoàn thành báo cáo đề tài **"Trực quan hóa PCA và ứng dụng PCA trong bộ dữ liệu Digit và khả năng dự đoán mắc bệnh ung thư vú",** nhóm em xin gửi lời cảm ơn chân thành nhất tới thầy **Phạm Đức Hồng**, đã truyền đạt, giảng dạy cho chúng em những kiến thức, những kinh nghiệm quý báu trong thời gian học tập và rèn luyện, tận tình hướng dẫn chúng em trong quá trình làm báo cáo này.

Nhóm em cũng gửi lời cảm ơn tới bạn bè đã đóng góp những ý kiến quý bàu để nhóm em có thể hoàn thành báo cáo tốt hơn. Tuy nhiên, do thời gian và trình độ có hạn nên báo cáo này chắc chắn không tránh khỏi những thiếu sót, nhóm em rất mong được sự đóng góp ý kiến của các thầy và toàn thể các bạn.

Một lần nữa, em xin chân thành cảm ơn và luôn mong nhận được sự đóng góp của tất cả mọi người.

Nhóm sinh viên thực hiện

Nguyễn Văn Nam

Hà Qúy Đức

Vũ Quang Huy

LỜI MỞ ĐẦU

Lý do chọn đề tài

Ngày nay, với sự phát triển mạnh mẽ của Công nghệ thông tin, các mô hình tự động hóa ngày càng được ứng dụng trong thực tế nhiều hơn. Song song với nó, khai thác dữ liệu để phục vụ trong công cuộc Cách mạng 4.0 là không thể thiếu. Dữ liệu trong thực tế thì vô cùng đa dạng. Muốn sử dụng dữ liệu một cách thông minh và có ích nhất, chúng ta cần quan tâm tới các đặc tính (feature) của dữ liệu. Chúng ta có thể quan sát được trong không gian 2 chiều, 3 chiều, nhưng dữ liệu thì lại có rất nhiều chiều. Làm sao để có thể trực quan hóa dữ liệu lên không gian 2 chiều, 3 chiều? Để trả lời câu này, chúng em xin chọn đề tài: **"Trực quan hóa PCA - ứng dụng PCA trong bộ dữ liệu Digits và dự đoán khả năng mắc bệnh ung thư vú"** để làm rõ.

Trong khuôn khổ bài tập lớn của nhóm, chúng em xin được trình bày giảm chiều dữ liệu bằng phương pháp phân tích thành phần chính (PCA) ứng dụng trong bộ dữ liệu Digits và dự đoán khả năng mắc bệnh ung thư vú.

Cấu trúc báo cáo bao gồm các chương như sau:

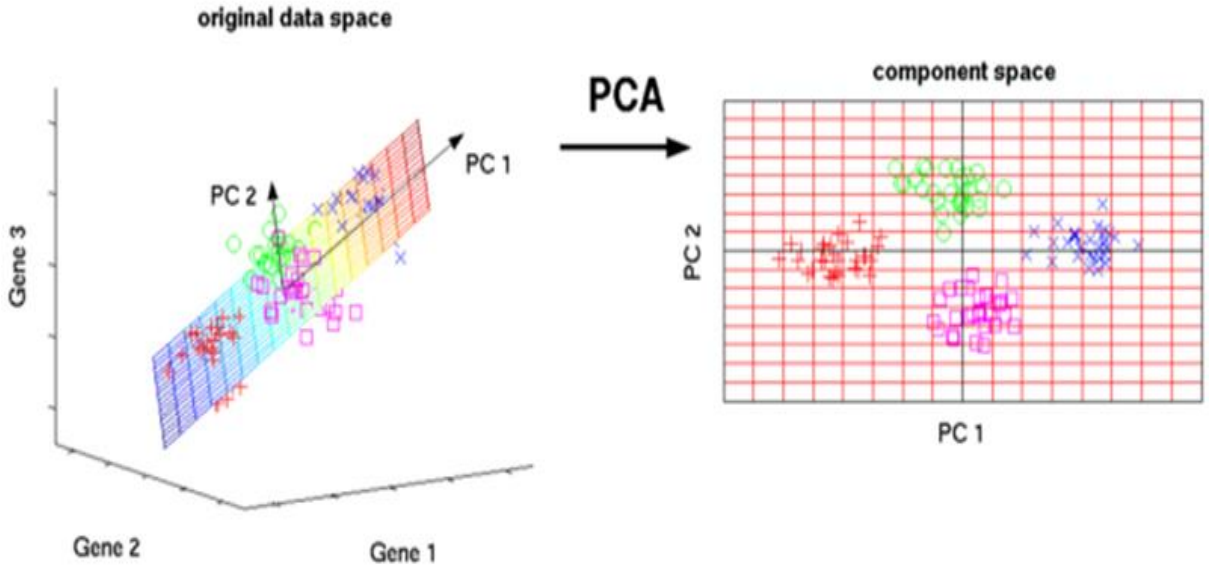
Chương 1: Giới thiệu về Phương pháp phân tích thành phần chính (PCA)

Chương 2: Cơ sở toán học trong PCA

Chương 3: Ứng dụng thuật toán PCA trong bộ dữ liệu Digits và dự đoán khả năng mắc bệnh ung thư vú.

CHƯƠNG 1: GIỚI THIỆU VỀ PHƯƠNG PHÁP PHÂN TÍCH THÀNH PHẦN CHÍNH (PCA)

1.1. Thuật toán PCA (Principal Component Analysis)



Hình 1.1: Hình ảnh đại diện cho phương pháp giảm chiều PCA

Thuật toán phân tích thành phần chính (Principal Components Analysis - PCA) là một thuật toán thống kê sử dụng phép biến đổi trực quan để biến đổi một tập hợp dữ liệu từ một không gian nhiều chiều sang một không gian mới ít chiều hơn (2 hoặc 3 chiều) nhằm tối ưu hóa việc thể hiện sự biến thiên của dữ liệu.

Ý tưởng chính của PCA là ánh xạ các đặc trưng **n** chiều thành **k** chiều. **k** chiều này là một đối tượng trực giao hoàn toán mới, còn được gọi là thành phần chính, là đối tượng k chiều được tái tạo lại trên cơ sở đối tượng n chiều ban đầu.

Công việc của PCA là tìm một cách tuần tự một tập các trục tọa độ mới có liên quan mật thiết đến bản thân dữ liệu. Trong số đó, lựa chọn trục tọa độ mới thứ hai là mặt phẳng trực giao với trục toạn độ đầu tiên để tối đa hóa phương sai và trục thứ ba giống với trục thứ nhất. Bằng phép loại suy, có thể thu được n trục tọa độ như vậy. Với trục tọa độ mới thu được theo cách này, chúng ta thấy rằng hầu hết phương sai được chưa trong k trục tọa độ đầu tiên và phương sai chứa

trong trục tọa độ sai gần như bằng 0. Do đó, chúng ta có thể bỏ qua các trục còn lại và chỉ giữ lại k trục đầu tiên chứa hầu hết các phương sai. Trên thực tế, điều này tương đương với việc chỉ giữ lại các đặc trưng chứa hầu hết phương sai và bỏ qua các kích thược đặc trưng chứa phương sai gần như bằng 0, để đạt được quá trình giảm kích thước cho các đối tượng dữ liệu.

Nói một cách ngắn gọn: Sử dụng ít chỉ số toàn diện hơn để đại diện cho nhiều loại thông tin khác nhau trong mỗi biến, phân tích thành phần chính và phân tích nhân tố thuộc loại thuật toán giảm chiều này.

1.2. Giảm chiều dữ liệu

Giảm chiều dữ liệu là sự biến đổi dữ liệu từ không gian nhiều chiều thành không gian ít chiều để biểu diễn ở dạng chiều thấp đồng thời giữ lại một số thuộc tính có ý nghĩa của dữ liệu gốc, có ý tưởng là gần với chiều nội tại.

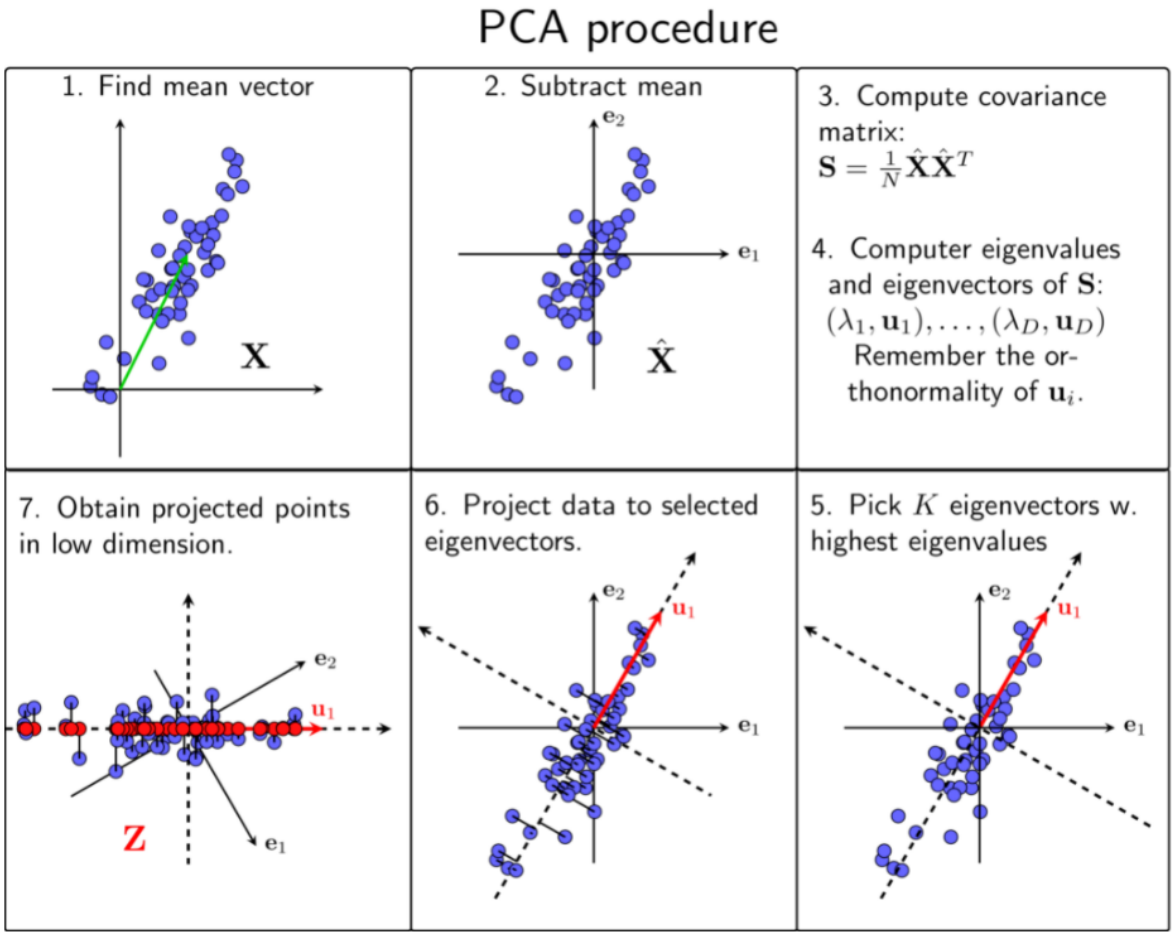
Phân tích dữ liệu trong không gian nhiều chiều có thể khó khăn vì nhiều lý do, dữ liệu thô có tính thưa thớt là một hậu quả của lời nguyền chiều và do đó việc phân tích trở lên khó tính toán, hơn nữa thuật toán có thể mất rất nhiều thời gian để xử lý dữ liệu. Giảm chiều dữ liệu là phổ biến trong các lĩnh vực có số lượng quan sát lớn hoặc số lượng biến lớn chẳng hạn như nhận dạng tiếng nói, tin học thần kinh và tin sinh học.

Tóm lại, giảm chiều là một phương pháp xử lý trước dữ liệu tính năng nhiều chiều. Giảm chiều là giữ lại các tính năng quan trọng nhất của dữ liệu, loại bỏ nhiễu và các tính năng không quan trọng, để đạt được mục đích cải thiện tốc độ xử lỹ dữ liệu.

Trong thực tế, sản xuất và ứng dụng, việc giảm chiều trong một phạm vi tổn thất thông tin nhất định có thể giúp chúng ta tiết kiệm rất nhiều thời gian và chi phí. Giảm chiều cũng đã trở thành một phương pháp tiền xử lý dữ liệu được sử dụng rất rộng rãi.

1.3. Các bước thực hiện thuật toán giảm chiều PCA

- Bước 1: Tính vector kỳ vọng của toàn bộ dữ liệu
- Bước 2: Trừ mỗi điểm dữ liệu đi vector kỳ vọng của toàn bộ dữ liệu
- Bước 3: Tính ma trận hiệp phương sai
- Bước 4: Tính các trị riêng và vector riêng của norm bằng một ma trận này, sắp xếp chúng theo thứ tự giảm dần của trị riêng.
- Bước 5: Chọn K vector riêng ứng với K trị riêng lớn nhất để xây dựng ma trân U_k có các cột tạo thành một hệ trực giao. K vector này còn được gọi là các thành phần chính tạo thành một không gian con gần với phân bố của dữ liệu ban đầu đã chuẩn hóa.
- Bước 6: Chiếu dữ liệu ban đầu đã chuẩn hóa xuống không gian con tìm được.
- Bước 7: Dữ liệu mới chính là tọa độ của các điểm dữ liệu trên không gian mới.



Hình 1.1: Các bước thực hiện PCA

1.4. Tiêu chí giảm chiều PCA

- Tái tạo gần nhất: Đối với tất cả các điểm trong tập mẫu, tổng sai số giữa điểm được tái tạo và điểm ban đầu là nhỏ nhất.
- Khả năng phân tách tối đa: Hình chiếu của mẫu trong không gian chiều thấp càng tách biệt càng tốt.

1.5. Ưu, nhược điểm của thuật toán PCA

1.5.1. Ưu điểm của thuật toán PCA

- Loại bỏ các đặc trưng tương quan (giảm các đặc trưng)
- Làm cho tập dữ liệu dễ sử dụng hơn.
- Cải thiện hiệu suất thuật toán.
- Giảm quá khớp (overfitting).
- Cải thiện trực quan hóa dữ liệu (dễ trực quan hóa khi có ít chiều)

1.5.2. Nhược điểm của thuật toán PCA

- Nếu người sử dụng đã có kiến thức nhất định về đối tượng quan sát và nắm vững một số đặc điểm của dữ liệu nhưng không thể can thiệp vào quá trình xử lý thông qua tham số hóa và các phương pháp khác thì có thể không đạt được hiệu quả mong đợi và hiệu quả không cao;
- Phân rã Eigenvalue có một số hạn chế, ví dụ, ma trận được biến đổi phải là ma trận vuông;
- Trong trường hợp phân bố không theo Gaussian, các thành phần chính thu được bằng phương pháp PCA có thể không tối ưu.
- Các biến độc lập trở nên khó hiểu hơn.
- Chuẩn hóa dữ liệu trước khi sử dụng PCA.
- Mất thông tin.

1.6. Ứng dụng thuật toán PCA

- Khám phá và trực quan hóa các tập dữ liệu nhiều chiều.
- Nén dữ liệu.
- Tiền xử lý dữ liệu.
- Phân tích và xử lý hình ảnh, giọng nói và giao tiếp.

- Giảm kích thước (quan trọng nhất), loại bỏ dư thừa dữ liệu và nhiễu.
- PCA trong nhận dạng ảnh nư nhận dạng khuôn mặt, …
- ứng dụng PCA trong phân tích mô tả định lượng
- Nếu ta có thể giảm chiều về 2 hoặc 3 chiều ta có thể dùng các loại đồ thị để hiểu thêm về dữ liệu mà ta đang có giúp dễ trực quan hơn.
- Xử lý vấn đề tương quan giữa các biến trong dữ liệu ban đầu bằng cách sử dụng biến mới trong không gian mà phương pháp PCA tìm được để mô tả dữ liệu.

CHƯƠNG 2: CƠ SỞ TOÁN HỌC SỬ DỤNG TRONG PRINCIPAL COMPONENT ANALYSIS – PCA

2.1. Độ lệch chuẩn (Standard Deviation)

- **Ý nghĩa:** đó tính biến động của giá trị mang tính thống kê. Nó cho thấy sự chênh lệch về giá trị của từng thời điểm đánh giá so với giá trị trung bình.
- **Biểu diễn toán học:**

$$\sigma = s = E\{X(t) - m_x(t)\}$$

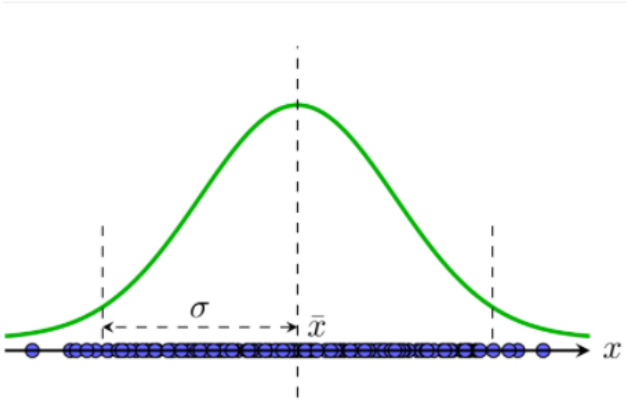
2.2. Kỳ vọng và ma trận hiệp phương sai

2.2.1. Dữ liệu một chiều

- Cho N giá trị từ x_1 đến x_N thì kỳ vọng và phương sai của bộ dữ liệu này được định nghĩa là:

$$\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n = \frac{1}{N}\mathbf{X1}\ \sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})^2$$

- Với 1 thuộc R^N là vector cột chứa toàn bộ phần từ 1, Kỳ vọng đơn giản là trung bình công của toàn bộ các giá trị.
- Phương sai là trung bình cộng của bình phương khoảng cách từ mỗi điểm tới kỳ vọng, phương sai càng nhỏ thi các điểm dữ liệu càng gần với kỳ vọng, tức là các điểm dữ liệu càng giống nhau, phương sai càng lớn thì ta nói dữ liệu càng có tính phân tán.



Hình 2.1: Ví dụ về kỳ vọng và phương sai trong không gian một chiều

2.2.2. Dữ liệu nhiều chiều

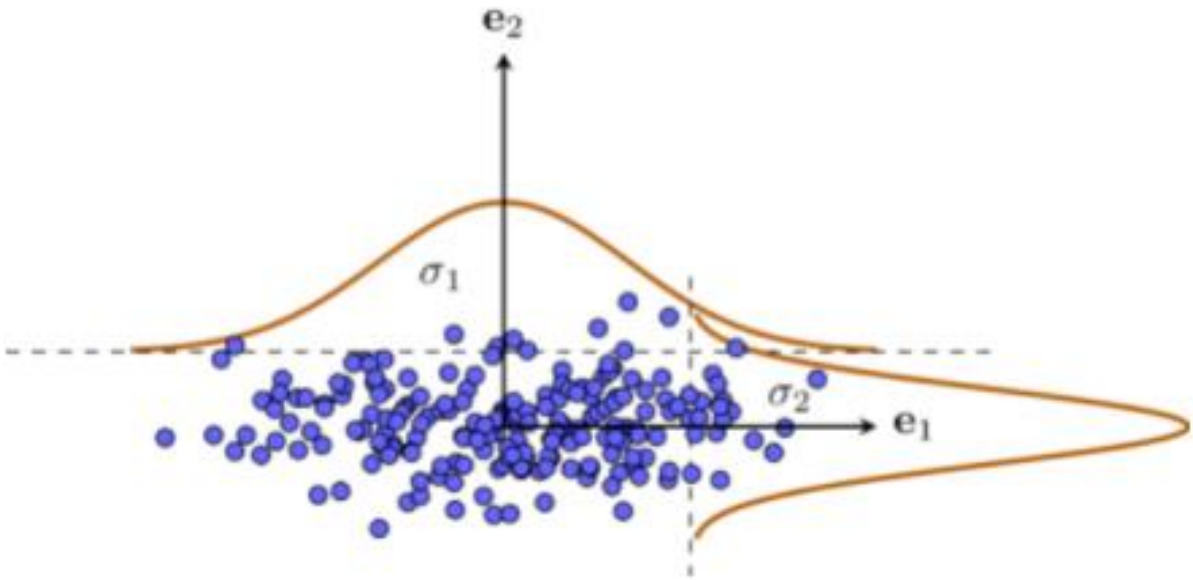
- Cho N điểm dữ liệu được biểu diễn bởi các vector cột x_1 đến x_N khi đó vector kỳ vọng và ma trận hiệp phương sai của toàn bộ dữ liệu được định nghĩa là:

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n \quad \mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T = \frac{1}{N}\hat{\mathbf{X}}\hat{\mathbf{X}}^T$$

Trong đó $\hat{\mathbf{X}}$ được tạo bằng cách trừ mỗi cột của $\mathbf{X}$ đi $\bar{\mathbf{x}}$:

$$\hat{\mathbf{x}}_n = \mathbf{x}_n - \bar{\mathbf{x}}$$

- Các công thức này khá tương đồng với với các công thức của dữ liệu một chiều, cho nên có một vài lưu ý như sau:
  - Ma trận hiệp phương sai là ma trận đối xứng hơn thế nữa nó là một ma trận nửa xác định dương.
  - Mọi phần tử trên đường chéo của ma trận hiệp phương sai là các số không âm, chúng cũng chính là phương sai của từng chiều của dữ liệu.
  - Nếu ma trận hiệp phương sai là ma trận đường chéo, ta có dữ liệu hoàn toàn không tương quan giữa các chiều.



Hình 2.2: Dữ liệu trên không gian hai chiều không tương quan

CHƯƠNG 3: ỨNG DỤNG TRỰC QUAN HÓA PCA TRONG BỘ DỮ LIỆU DIGITS VÀ DỰ ĐOÁN KHẢ NĂNG MẮC BỆNH UNG THƯ VÚ

3.1. Mô tả bài toán

3.1.1. Mô tả bài toán trực quan hóa PCA trong bộ dữ liệu Digits

Trực quan hóa PCA là bài toán giải thích về phân tích thành phần chính ứng dụng trong nhiều bài toán như nhận dạng chữ viết, nhận dạng khuôn mặt, nhận dạng ngôn ngữ ký hiệu số…

Vấn đề nhận dạng ngôn ngữ ký hiệu số là một thách thức lớn đối với các nhà nghiên cứu ngày trước. Bài toán lớn luôn đặt ra phía trước vì sự phức tạp của việc nhận dạng ngôn ngữ ký hiệu số. Do vậy, xây dựng hệ thống nhận dạng ngôn ngữ ký hiệu số có thể nhận dạng đuược bất cứ hành động cử chỉ nào của bàn tay một cách dễ dàng. Chúng tôi sẽ sử dụng trực quan hóa PCA để biểu diễn bài toán.

3.1.2. Mô tả bài toán dự đoán khả năng mắc bệnh ung thư vú

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 hớn. 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.

- Input: Thông tin, đặc tính của người có khả năng mắc bệnh hoặc không.
- Ouput: Kết quả người được chuẩn đoán có bị mắc bệnh hay không.

3.2. Môi trường thực nghiệm

Hình 3.1: 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.

3.3. Xây dựng bộ dữ liệu

3.3.1. Bộ dữ liệu cho bài toàn trực quan hóa PCA với bộ dữ liệu Digits

- Chúng tôi sử dụng tập dữ liệu Digits, thường dùng để đánh giá hiệu quả của giải thuật nhận dạng ký hiệu ngôn ngữ số. Tập dữ liệu Digits có nguồn gốc Thổ Nhĩ Kỳ học sinh trung học Ankara Ayrancu Anadolu.
- Tập dữ liệu được chuẩn bị bởi nhóm học sinh:

Hình 3.2: Tập dữ liệu Digits

- Chi tiết bộ dữ liệu:
  - Kích thước hình ảnh: 64 x 64 pixel
  - Không gian màu: GGB
  - Số lớp: 10 (Chữ số : 0 - 9)
  - Số lượng sinh viên tham gia: 218
  - Số lượng mẫu cho mỗi học sinh: 10

3.3.2. 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 3.3: Bộ dữ liệu dự đoán khả năng mắc bệnh ung thư vú

3.4. Áp dụng phân loại MLPClassifier cho bài toán Bộ dữ liệu chữ số ngôn ngữ ký hiệu với PCA

3.4.1. Cài đặt thuật toán

- Khai báo các thư viện cần thiết cho bài toán.

```
'''Khai báo thư viện cần thiết cho bài toán'''
import numpy as np
import matplotlib.pyplot as plt
import time
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.decomposition import PCA
from sklearn.metrics import classification_report
import warnings
warnings.filterwarnings('ignore')
```

- Đọc dữ liệu Digits

```
'''load dữ liệu cho bài toán'''
X = np.load("../Dataset/DigitsData/X.npy")
Y = np.load("../Dataset/DigitsData/Y.npy")
print("Data shape: ", X.shape)
print(X.shape[0], "sample, ", X.shape[1], "x", X.shape[2], 'size grayscall image.\n')
print("Labels shape: ", Y.shape)
```

- Hiển thị một số mẫu dữ liệu

```
'''Hiển thị một số dữ liệu mẫu'''
print('Hiển thị một số mẫu:')
img_size = 64
n = 10
plt.figure(figsize=(20, 4))
image_index_list = [260, 900, 1800, 1600, 1400, 2061, 700, 500, 1111, 100]
for i in range(1, n + 1):
    ax = plt.subplot(1, n, i)
    plt.imshow(X[image_index_list[i - 1]].reshape(img_size, img_size))
    # plt.gray()
    plt.axis('off')
    title = "Sign " + str(i - 1)
    plt.title(title)
plt.show()
```

- Chia tập dữ liệu theo tỉ lệ 8:2

```
'''Chia tập dữ liệu thành tập dữ liệu huấn luyện và dữ liệu thử nghiệm (train và test)'''
X_fat = np.array(X).reshape(2062, 64*64)
X_train, X_test, Y_train, Y_test = train_test_split(X_fat, Y, test_size=0.2, random_state=42)
print('Training shape:', X_train.shape)
print(X_train.shape[0], 'sample,', X_train.shape[1], 'size grayscale image.\n')
print('Test shape:', X_test.shape)
print(X_test.shape[0], 'sample,', X_test.shape[1],'size grayscale image.\n')
```

- Chạy mô hình học máy với tập dữ liệu gốc

```
'''Sử dụng mô hình MLPClasifier khi chưa sử dụng giảm chiều PCA để huấn luyện'''
clf = MLPClassifier(solver='adam', alpha=1e-5, hidden_layer_sizes=(100, 100, 100, 100), random_state=1)


start = time.time()
clf.fit(X_train, Y_train)
end = time.time()
print("Training time is " + str(end - start) + " second.")
'''Dự đoán mô hình khi chưa sử dụng phương pháp giảm chiều PCA'''
y_hat = clf.predict(X_test)
print("{}: {:.2f}%".format("Accuracy", accuracy_score(Y_test, y_hat)*100))
```

- Tính ra số chiều sau khi đã giảm chiều

```
'''Tính ra số chiều sau khi đã giảm chiều'''
pca_dims = PCA()
pca_dims.fit(X_train)
cumsum = np.cumsum(pca_dims.explained_variance_ratio_)
d = np.argmax(cumsum >= 0.7) + 1
```

- Tính toán giảm chiều và hiển thị ra hình ảnh trước khi giảm chiều và sau khi giảm chiều

```
pca = PCA(n_components= d)
X_reduced = pca.fit_transform(X_train)
X_recovered = pca.inverse_transform(X_reduced)
print("reduced shape: " + str(X_reduced.shape))
print("recovered shape: " + str(X_recovered.shape))
```

```
'''Hiển thị hình ảnh trước và sau khi giảm chiều PCA'''
f = plt.figure()
f.add_subplot(1, 2, 1)
plt.title("Trước khi giảm chiều")
plt.imshow(X_train[4].reshape((img_size, img_size)))
f.add_subplot(1, 2, 2)
plt.title("Sau khi giảm chiều")
plt.imshow(X_recovered[4].reshape((img_size, img_size)))
plt.show(block = True)
```

- Chạy mô hình học máy với tập dữ liệu đã sử dụng phương pháp giảm chiều PCA.

```
'''Sử dụng mô hình MLPClasifier khi sử dụng giảm chiều PCA để huấn luyện'''
clf_PCA = MLPClassifier(solver='adam', alpha=1e-5,
            hidden_layer_sizes= (100, 100, 100, 100), random_state=1)
start_PCA = time.time()
clf_PCA.fit(X_reduced, Y_train)
end_PCA = time.time()
print("Training time is " + str(end_PCA - start_PCA) + " second using PCA")
```

- Đánh giá mô hình học máy dự trên kết quả dự đoán với độ đo đơn giản Accuracy, Precisionm Recall

```
'''Đánh giá mô hình học dựa trên kết quả dự đoán
    (với độ đo đơn giản Accuracy, Precision, Recall)'''
y_hat_PCA_classes = np.argmax(y_hat_PCA, axis=1)
y_true = np.argmax(Y_test, axis=1)
confusion_mtx = confusion_matrix(y_true, y_hat_PCA_classes)
print("Ma trận dự doán:\n", confusion_mtx)
print("{}: {:.2f}%".format("Accuracy Score: ",
            accuracy_score(y_true, y_hat_PCA_classes)*100))
print(classification_report(y_true, y_hat_PCA_classes))
```

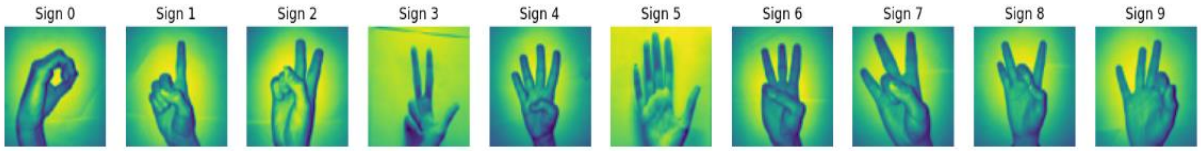
- Vẽ ma trận dự đoán

```
'''Vẽ ma trận dự đoán'''
f, ax = plt.subplots(figsize=(8, 8))
sns.heatmap(confusion_mtx, annot=True, linewidths=0.01,cmap="BuPu",
            linecolor="gray", fmt= '.1f',ax=ax)
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.title("Confusion Matrix")
plt.show()
```

3.4.2. Kết quả thực nghiệm

- Hiển thị một số thông tin cũng như một số mẫu dữ liệu

```
Data shape:  (2062, 64, 64)
2062 sample,  64 x 64 size grayscall image.

Labels shape:  (2062, 10)
Hiển thị một số mẫu:
```



- Số lượng dữ liệu sau khi chia tập train test

```
Training shape: (1649, 4096)
1649 sample, 4096 size grayscale image.

Test shape: (413, 4096)
413 sample, 4096 size grayscale image.
```

- Hiển thị thời gian train dữ liệu khi chưa giảm chiều và kết quả thử nghiệm

```
Training time is 27.277154684066772 second.
Accuracy: 72.15%
```

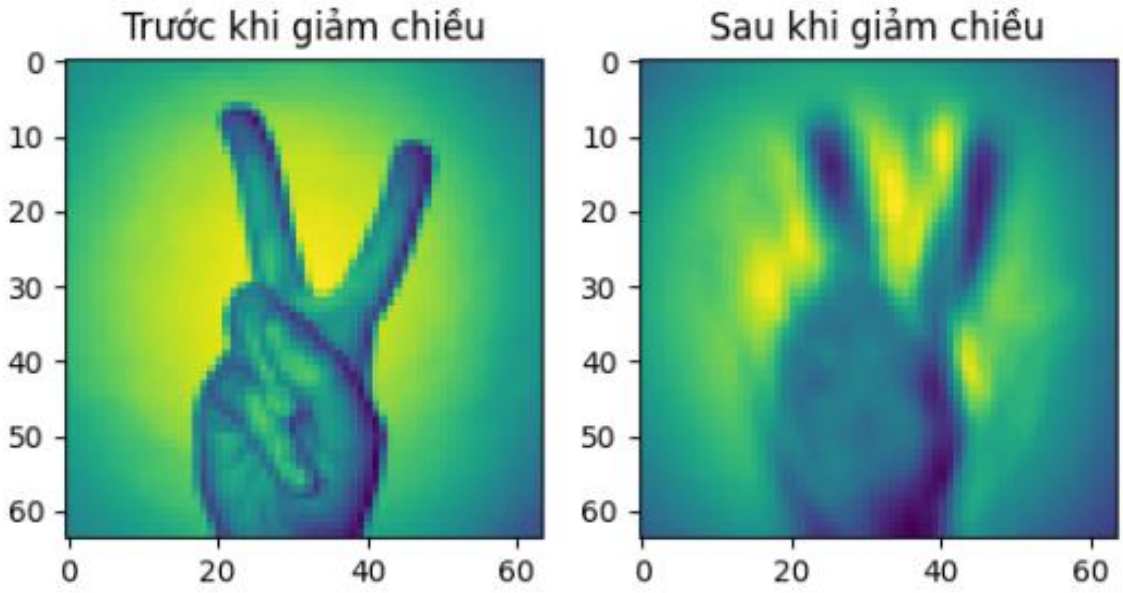
- Hiển thị số chiều sau khi giảm chiều và thời gian train dữ liệu khi sử dụng giảm chiều PCA

```
reduced shape: (1649, 40)
recovered shape: (1649, 4096)
Training time is 4.437165975570679 second using PCA
```

- Hiển kết quả dự đoán

```
Kết quả dự đoán:
 [[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 1]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]]
```

- Hiển thị hình ảnh trước và sau khi giảm chiều



- Hiển thị số lớp sau của bài toán

```
Số lớp:  [0 1 2 3 4 5 6 7 8 9]
```

- Hiển thị ma trận dự đoán

```
Ma trận dự đoán:
 [[38  0  0  0  0  0  1  1  0  0]
 [ 3 42  0  2  0  0  1  0  0  0]
 [ 4  0 44  1  0  1  0  0  0  0]
 [ 2  0  2 27  0  0  3  0  1  0]
 [ 1  1  0  1 29  0  0  0  0  0]
 [ 6  0  1  0  1 28  1  0  0  1]
 [ 2  1  2  4  0  1 29  0  0  2]
 [ 1  0  0  0  0  0  0 36  0  1]
 [ 3  0  4  1  2  0  0  0 35  0]
 [ 0  0  0  0  0  0  3  0  0 43]]
```

- Hiển thị đánh giá mô hình

```
Accuracy Score: : 84.99%
              precision    recall  f1-score   support

           0       0.63      0.95      0.76        40
           1       0.95      0.88      0.91        48
           2       0.83      0.88      0.85        50
           3       0.75      0.77      0.76        35
           4       0.91      0.91      0.91        32
           5       0.93      0.74      0.82        38
           6       0.76      0.71      0.73        41
           7       0.97      0.95      0.96        38
           8       0.97      0.78      0.86        45
           9       0.91      0.93      0.92        46

    accuracy                           0.85       413
   macro avg       0.86      0.85      0.85       413
weighted avg       0.87      0.85      0.85       413
```

- Hiển thị hình ảnh của ma trận dự đoán

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

3.5.1. 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
```

- 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()
```

3.5.2. 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
```

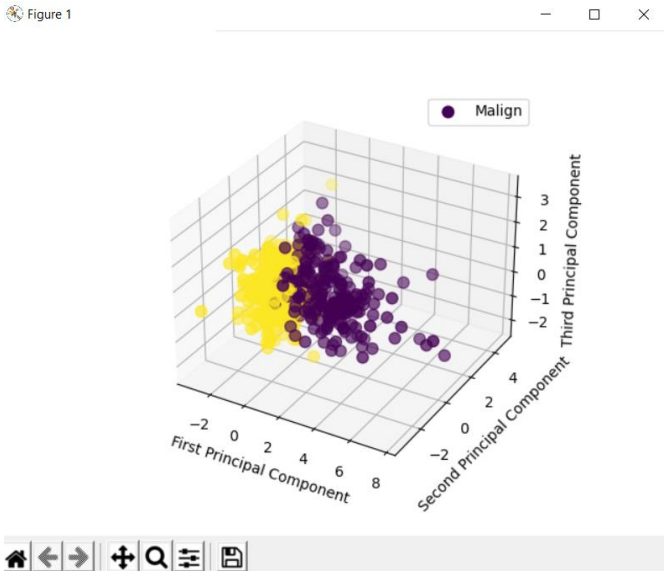
- Dữ liệu nhãn đầu ra Y:

```
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]]
```

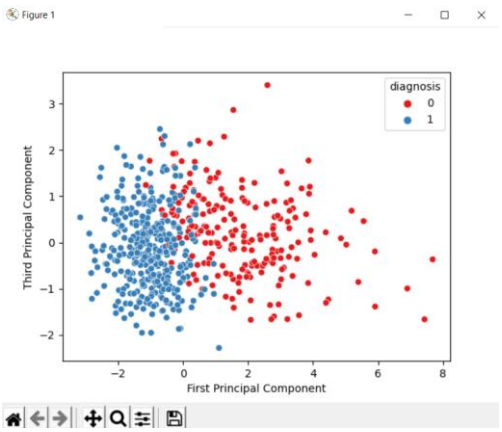
- Biểu diễn dữ liệu trên không gian 3D

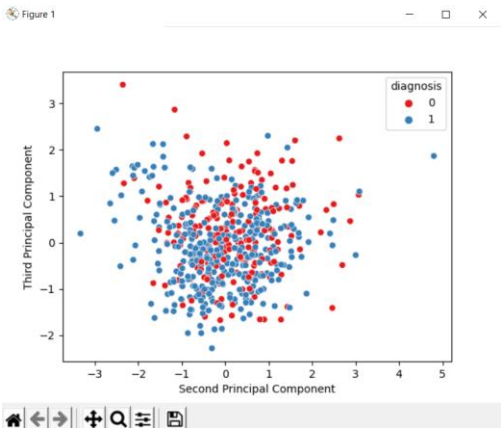


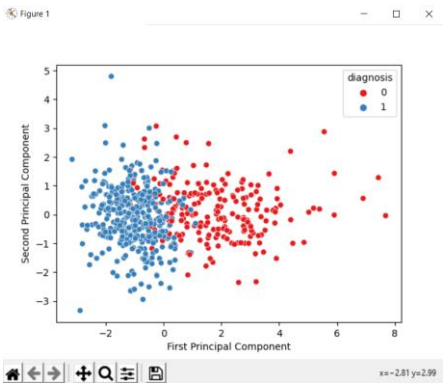
- Số chiều sau khi sử dụng phương pháp PCA để giảm chiều dữ liệu

```
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 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
```

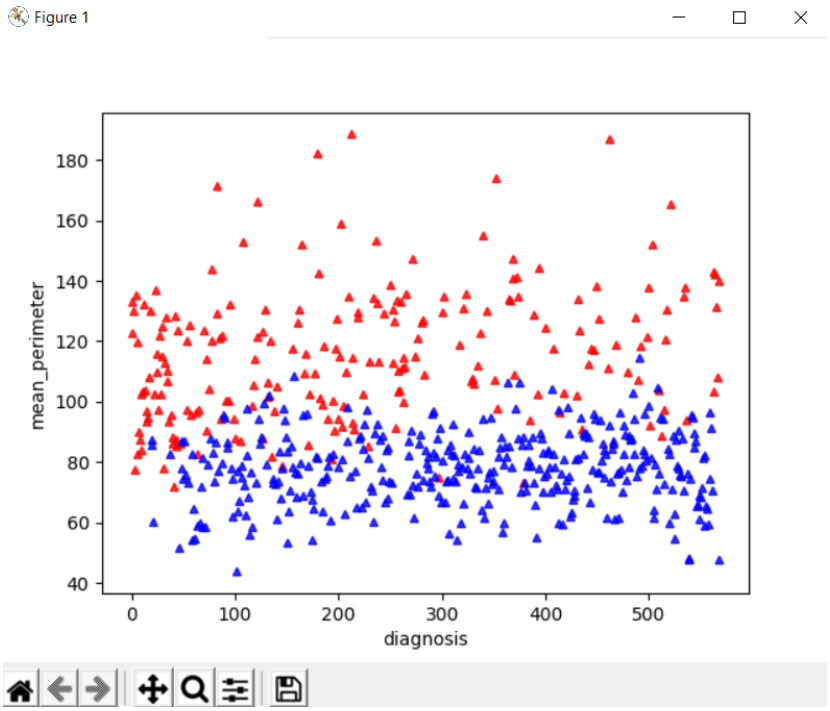
- Đối với Precision, Recall:

```
              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

Đối với dữ liệu nhiều chiều, phương pháp sử dụng thuật toán phân tích thành phần chính PCA cho kết khả quan, có ý nghĩa khoa học và giá trị thực tiễn. Tuy nhiên trong gia đoạn thử nghiệm nên các kết quả giảm chiều chưa được như mong đợi. Điều này do việc trích chọn đặc trưng cũng như việc lựa chọn các tham số phù hợp cho bài toán.

Trong thời gian tới, chúng em sẽ tiếp tục nâng cấp và hoàn thiện nhằm nâng cao tỉ lệ chính xác để giải quyết bài toán một cách nhanh gọn, tiết kiệm chi phí tối đa và dữ liệu được sử dụng một cách có ích.

TÀI LIỆU THAM KHẢO

[1] https://www.easy-tensorflow.com/tf-tutorials/linear-models/linear-classifier

[2] https://machinelearningcoban.com/2017/01/08/knn/

[3] https://github.com/thandongtb/tf_tutorial/blob/master/classification/mnist_softmax