# Printical Component Analysis – PCA

## Giới thiệu PCA

Printical Component Analysis là phép phân tích thành phần chính mục đích chính nhằm biến đổi cũng như giảm chiều (feature/attribute) tập dataset dựa vào phương sai và các hệ số tương qua giữa các chiều, tập dataset mới sẽ có số chiều ít hơn mang hầu hết các thông tin của tập dataset cũ.

## Giới thiệu Software Defect Prediction Dataset

Dataset với các số liệu của một file source code như số rẽ nhánh, số điều kiện, số dòng code, số comment, … để dự đoán file source code đó có deffective hay không.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên file:** | dataset\_for\_PCA\_LDA.csv | **Số datapoint (dòng):** | 124 |
| **Số attribute:** | 39 | **Số classtification:** | 2 (YES/NO) |

## Hiện thực PCA trên Software Defect Prediction Dataset

### ***Lấy dữ liệu và chuẩn hóa dataset***

****

Tách dataset thành X là tập 124 datapoint, y là target, đồng thời dùng LabelEncoder để chuyển label dạng chuỗi thành số tức YES/NO thành 1/0

Sử dụng thư viện StandardScaler để chuẩn hóa data



### ***Sử dụng thuật toán Logistic Regression***

Lý do chúng ta sử dụng Logistic Regression để phân lóp là do các attribute của Software Defect Prediction có giá trị liên tục (thuộc tập số thực R)



Chia tập dataset trên với 20% để testing 80% để training, sau đó dùng GaussiNB của Scikitlearn để training và dự đoán cho ra kết quả y\_predict

Bây giờ chúng ta sẽ xuất các thông số đánh giá thử model



Confusion matrix và các thông số khác như accuracy, precision, recall và F1 như sau



Trong 20% testing data (X\_test, y\_test) tức 25 datapoint model Linear Regression đã đoán đúng 6 YES và 12 NO.

### ***Đánh giá trước khi sử dụng PCA***

Tính toán và in ma trận hiệp phương sai cov\_mat với ma trận hệ số tương quan cor\_mat





Tính toán các trị riêng eig\_vals, các vector riêng eig\_vecs theo từng cặp, sau đó sắp xếp chúng theo thứ tự giảm giần của trị riêng.





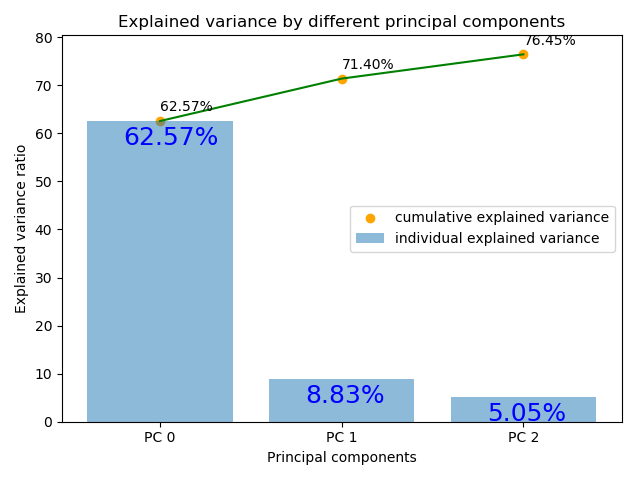
Bây giờ tính tỉ lệ của các PC xem chúng chứa bao nhiêu thông tin cũ





Có thể thấy với 3 PC đầu tiên đã chứa tới 76.45% thông tin từ dataset ban đầu. Vẽ thử đồ thị biễu diễn tỉ lệ thông tin mà các PC này mang.





### ***Áp dụng PCA***

Ghép 3 vector riêng chúng ta đã chọn bên trên thành matrix\_w



Sau đó chiếu hết datapoint ban đầu xuống không gian mới 3 chiều này



Nếu sử dụng thư viện chúng ta không cần làm 2 việc trên mà chỉ cần



Xong vậy là chúng ta có X\_new và y sử dụng lại Linear Regression và đánh giá lần 2.

### ***Sử dụng lại thuật toán Linear Regression và đánh giá***

Vì tính khách quan chúng ta không nên dùng train\_test\_split trên X\_new vì nó sẽ trộn (shuffle) và chọn ngẫu nhiên để chia lại cho training/testing data gây ra sự khác biệt ở lần đầu sử dụng Linear Regression. Hãy sử dụng matrix\_w để chiếu training/testing data cũ thành mới



Phân lớp lại với Linear Regression

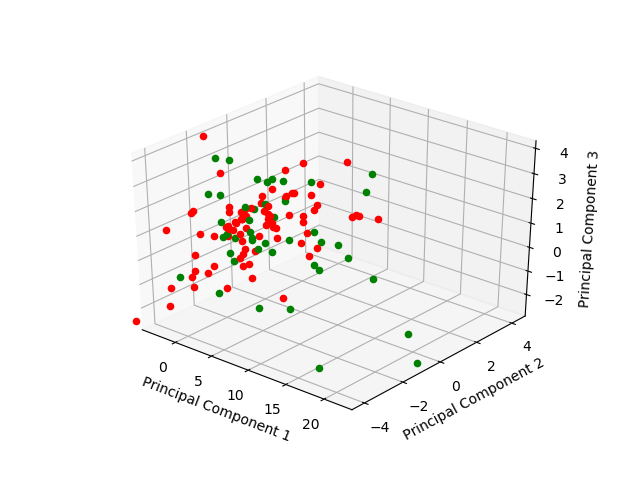




## Kết luận

Theo confusion matrix thì các thông số giảm đôi chút đơn giản là tập dataset mới chiều đã giảm chỉ mang 76.45% thông tin cũ nhưng nhờ áp dụng PCA chúng ta có thể giảm/bỏ đi các attribute không cần thiết. Vẽ thử tập dataset mới





Có thể thấy đây là đặc tính của PCA sự tách biệt giữa các class không rõ ràng.

# Linear Discriminant Analysis – LDA

## Giới thiệu LDA

Linear Discriminant Analysis là phương pháp giảm chiều dữ liệu đồng thời cũng là phương pháp phân lớp. LDA là cho tập dataset mới sẽ có số chiều ít hơn và mỗi datapoint mang đặc trưng của class đó.

## Hiện thực LDA trên Software Defect Prediction Dataset

### ***Lấy dữ liệu, chuẩn hóa dataset và sử dụng thuật toán Logistic Regression***

Tương tự ở phần PCA trên

### ***Áp dụng LDA***

Đầu tiên chúng ta sẽ tính mean vector



Chỉ có 2 lable nên sẽ có 2 mean vector



Kế tiếp sẽ tính ma trận Within-class scatter S\_W





Và ma trận Between-class scatter S\_B





Xong vậy là chúng ta có X\_new và y sử dụng lại Linear Regression và đánh giá lần 2.

### ***Sử dụng lại thuật toán Linear Regression và đánh giá***

Vì tính khách quan chúng ta không nên dùng train\_test\_split trên X\_new vì nó sẽ trộn (shuffle) và chọn ngẫu nhiên để chia lại cho training/testing data gây ra sự khác biệt ở lần đầu sử dụng Linear Regression. Hãy sử dụng matrix\_w để chiếu training/testing data cũ thành mới



Phân lớp lại với Linear Regression





## Kết luận

# Multiple Linear Regression

# Polymial Linear Regression

# Logistic Linear Regresssion