Họ Tên Và MSSV:

Đỗ Thành Huy – SE150235

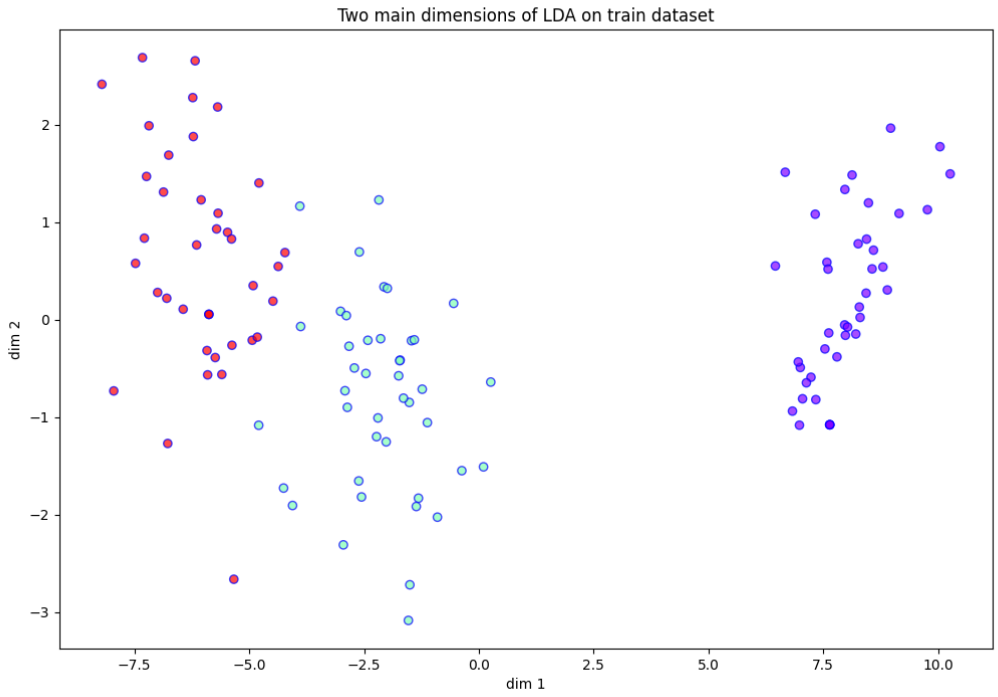
MINI PROJECT MAI391

I. Nội Dung

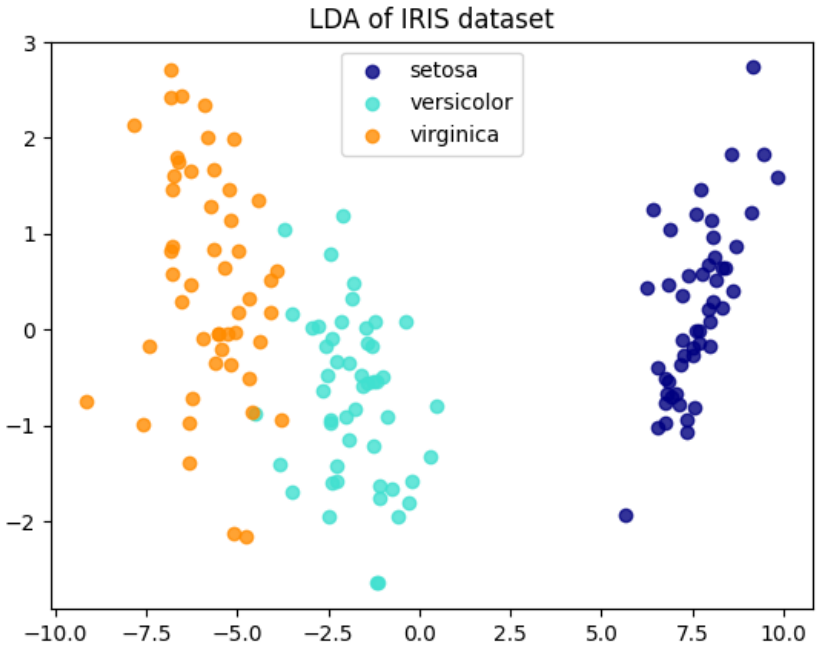
1. Giới Thiệu Và Mục Tiêu Nghiên Cứu :

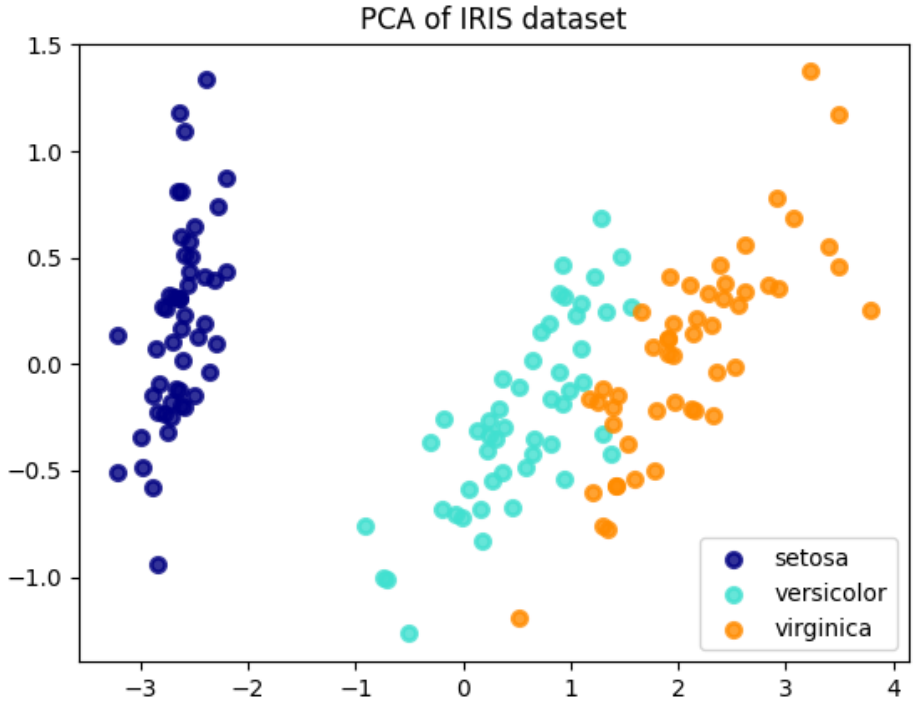
- Lý do nghiên cứu vì muốn hiểu rõ thuật toán LDA, cách vận hành của chúng trong các bài toán, hiểu tương quan về các tham số của LDA vận hành.
- LDA cũng liên quan chặt chẽ đến PCA và phân tích nhân tố ở chỗ cả hai đều tìm kiếm các tổ hợp tuyến tính của các biến giải thích tốt nhất cho dữ liệu. LDA cố gắng mô hình hóa sự khác biệt giữa các lớp dữ liệu một cách rõ ràng. Ngược lại, PCA không tính đến bất kỳ sự khác biệt nào về lớp và phân tích nhân tố xây dựng các kết hợp tính năng dựa trên sự khác biệt hơn là tương đồng. Phân tích phân biệt cũng khác với phân tích nhân tố ở chỗ nó không phải là một kỹ thuật phụ thuộc lẫn nhau: phải thực hiện sự phân biệt giữa các biến độc lập và các biến phụ thuộc (còn gọi là biến tiêu chí).
- LDA hoạt động khi các phép đo được thực hiện trên các biến độc lập cho mỗi lần quan sát là các đại lượng liên tục. Khi xử lý các biến độc lập phân loại, kỹ thuật tương đương là phân tích tương ứng phân biệt.
- Phân tích phân biệt được sử dụng khi các nhóm được biết trước (không giống như trong phân tích cụm). Mỗi trường hợp phải có điểm cho một hoặc nhiều biện pháp dự báo định lượng, và điểm cho một biện pháp nhóm. Nói một cách dễ hiểu, phân tích chức năng phân biệt là phân loại - hành động phân phối mọi thứ thành các nhóm, lớp hoặc danh mục cùng loại.
- Mục đích chính của bài báo này là quá trình và kết quả nghiên cứu chính của thuật toán LDA:

Kết quả cho ra bảng biểu diễn dữ liệu dataset iris đã dùng khi sử dụng thuật toán LDA:



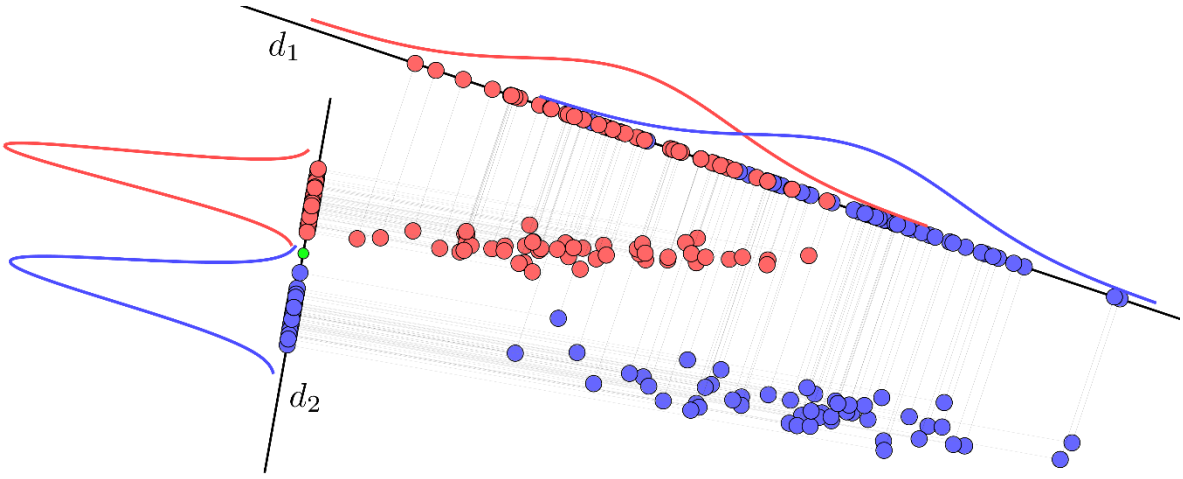
So sánh 2 thuật Toán LDA và PCA:





2. Lý Thuyết:

- Trong quá trình học tập thì chúng ta đã làm quen với thuật toán PCA, thuật toán này giúp giảm chiều dữ liệu nhưng vẫn giữ được cấu trúc biến động của bộ dữ liệu. có thể xem PCA là một thuật toán thuộc lớp mô hình học không giám sát vì việc lựa chọn các thành phần chính không phụ thuộc vào nhãn của biến mục tiêu. Trong một số bộ dữ liệu của học có giám sát, việc chiếu theo thành phần chính có thể không giúp cho việc phân loại tốt hơn so với thành phần phụ. Chẳng hạn như hình minh họa bên dưới:



Hình ảnh minh họa bộ dữ liệu gồm hai nhãn tương ứng với các chấm tròn xanh và đỏ. Giả định rằng thuật toán PCA tìm được hai véc tơ chiếu là d1 và d2. Khi đó d1 sẽ là thành phần chính vì thông tin về độ biến động của dữ liệu được giữ lại là nhiều nhất và d2 sẽ là thành phần phụ. Tuy nhiên đối với việc phân loại thì phương chiếu d2 có thể hoàn toàn tách rời các lớp trong khi d1 thì không.

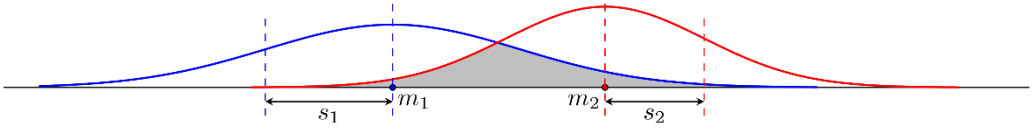
- Như vậy ví dụ trên đã cho thấy việc lựa chọn thành phần chính không phải lúc nào cũng là tốt nhất trong bài toán phân loại của học có giám sát. LDA là viết tắt của cụm từ Linear Discriminant Analysis (đôi khi còn được gọi là thuật toán Normal Discriminant Analysis hoặc Discriminant Function Analysis). Thuật toán này có thể khắc phục được các nhược điểm trên bằng cách kế thừa ý tưởng giảm chiều dữ liệu mà vẫn đảm bảo phân loại tốt các lớp của bộ dữ liệu.
- Trong bài nghiên cứu, thì LDA được sử dụng trong các mô hình sau đây:

a. Thuật Toán LDA trong phân loại nhị phân:

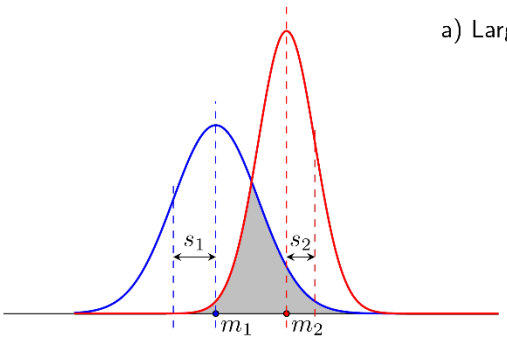
- Xét thuật toán LDA trên bài toán phân loại nhị phân. Giả sử dữ liệu đầu vào là tập hợp các điểm $\{(xi,yi)\}^{N}_{i=1}$, trong đó $x_i \in R^D$ và $y_i \in \{0,1\}$. Giả sử rằng các điểm nhãn 0 tuân theo phân phối chuẩn $N(m_1,s^2_1)$ (m_1 và s^2_1 lần lượt là trung bình và phương sai). Trong khi các điểm nhãn 1 tuân theo phân phối $N(m_2,s^2_2)$. Sau khi thực hiện phép chiếu tuyến tính từ một điểm dữ liệu x_i xuống trục chính ta thu được một số vô hướng z_i. Tọa độ của z_i có thể được mô tả bởi tích vô hướng với một véc tơ w:

$$zi=w^T x_i$$

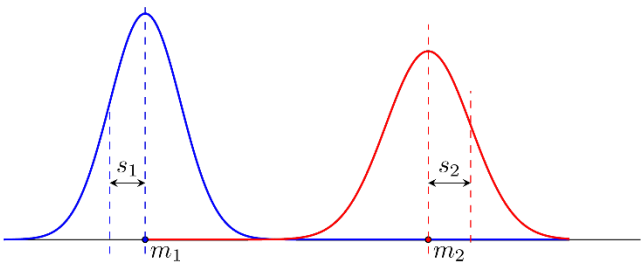
- Trọng số w cũng chính là giá trị mà thuật toán cần tìm kiếm trong bài toán tối ưu được trình bày bên dưới.
- Trên trục chính, chúng ta kì vọng rằng các lớp có khoảng cách tâm là lớn nhất và bán kính trên từng lớp là nhỏ nhất.



a) Large $(m_1 - m_2)^2$, large $s_1^2 + s_2^2$



b) Small $(m_1 - m_2)^2$, small $s_1^2 + s_2^2$



c) Large $(m_1 - m_2)^2$, small $s_1^2 + s_2^2$

- Hình a tương ứng với: khoảng cách tâm lớn, bán kính cụm lớn. Hình b: khoảng cách tâm nhỏ, bán kính cụm nhỏ. Hình c: khoảng cách tâm lớn, bán

kính cụm nhỏ. Mức độ dễ phân loại giữa hai lớp theo thứ tự lần lượt là c > b > a.

- Khoảng cách tâm được đo lường thông qua phương sai giữa các cụm (between class variance):

$$(m_1 - m_2)^2 = (\frac{1}{N_1}\sum_{i \in C_1} z_i - \frac{1}{N_2}\sum_{j \in C_2} z_j)^2 = ||\mathbf{w}^\intercal(\mathbf{m}_1 - \mathbf{m}_2)||_2^2$$

- Trong khi đó bán kính được đo lường thông qua phương sai trong từng cụm (within class variance):

$$s_k^2 = \sum_{n \in C_k}(z_n - m_k)^2, \quad k = 1, 2$$

- Thuật toán LDA đối với bài toán phân loại nhị phân xây dựng một hàm mục tiêu dạng:

$$J(\mathbf{w}) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2}$$

- Bài toán tối ưu sẽ trở thành bài toán tối đa hóa hàm mục tiêu. Nghiệm của bài toán:

$$\mathbf{w} = \arg\max_{\mathbf{w}} J(\mathbf{w})$$

b. LDA với phân loại đa lớp và không gian chiếu đa chiều:

- Trong trường hợp tổng quát giả sử từ dữ liệu đầu vào là $X \in R^{N \times D}$ trong đó N là số quan sát và D là số chiều gốc. Chúng ta cần giảm dữ liệu này về ma trận $Z \in R^{N \times K}$ với K<<D. Khi đó mỗi một quan sát $z_i \in R^K$ là hình chiếu của xi thông qua phép nhân ma trận:

$$z_i = Wx_i$$

- Với $W \in R^{K \times D}$. Đây cũng chính là ma trận mà chúng ta cần tìm kiếm trong bài toán phân loại đa lớp.

Từ đó suy ra:

$$Z^\intercal = WX^\intercal$$

- Ở đây ta chuyển vị ma trận để mỗi quan sát là một dòng trở thành cột.
- Trong trường hợp phân loại đa lớp, để so sánh khoảng cách giữa các tâm thì chúng ta sử dụng phương sai tâm. Tức là giả sử chúng ta có C nhãn cần phân chia với tâm là $m_k \in R^D, \forall k = \overline{1, C}$ và số lượng quan sát là N_k. Hình chiếu của các tâm này lên không gian giảm chiều là $e_k = Wm_k \in R^K$. Khi đó:
- Phương sai tâm chính là :

$$
\begin{aligned}
\mathbf{S}_{\text{between}} &= \sum_{k=1}^{C} N_k ||\mathbf{e}_k - \bar{\mathbf{e}}||_2^2 \\
&= \sum_{k=1}^{C} N_k ||\mathbf{W}(\mathbf{m}_k - \bar{\mathbf{m}})||_2^2 \\
&= \sum_{k=1}^{C} N_k \text{trace}\left(\mathbf{W}^\intercal \underbrace{(\mathbf{m}_k - \bar{\mathbf{m}})^\intercal (\mathbf{m}_k - \bar{\mathbf{m}})}_{\mathbf{S}_{Bk}} \mathbf{W} \right) \\
&= \text{trace}\left(\mathbf{W}^\intercal \left[N_k \sum_{k=1}^{C} \mathbf{S}_{Bk} \right] \mathbf{W} \right) \\
&= \text{trace}\left(\mathbf{W}^\intercal \mathbf{S}_B \mathbf{W} \right)
\end{aligned}
$$

- Dòng thứ 2 suy ra dòng thứ 3 là vì theo công thức: $||a||^2_2=\text{trace}(a^\intercal a)$ và $||A||^2_F=\text{trace}(A^\intercal A)$.
- Phương sai trên từng cụm vẫn không thay đổi so với trường hợp phân loại nhị phân:

$$
\begin{aligned}
s_k^2 &= \sum_{i \in C_k} ||\mathbf{z}_i - \mathbf{e}_k||_2^2 = ||\mathbf{Z}_k - \mathbf{E}_k||_F^2 \\
&= ||\mathbf{W}^\intercal (\mathbf{X}_k - \mathbf{M}_k)||_F^2 \\
&= \text{trace}\left(\mathbf{W}^\intercal \underbrace{(\mathbf{X}_k - \mathbf{M}_k)(\mathbf{X}_k - \mathbf{M}_k)^\intercal}_{\mathbf{S}_{Wk}} \mathbf{W} \right) \\
&= \text{trace}\left(\mathbf{W}^\intercal \mathbf{S}_{Wk} \mathbf{W} \right)
\end{aligned}
$$

- Ở đây, E_k,M_k lần lượt là những ma trận có các dòng lặp lại của véc tơ tâm e_k và m_k.
- Như vậy tổng phương sai trên từng cụm sẽ là:

$$
\begin{aligned}
\mathbf{S}_{\text{within}} &= \sum_{i=1}^{C} s_k^2 \\
&= \text{trace}\left(\mathbf{W}^\intercal \left[\sum_{k=1}^{C} \mathbf{S}_{Wk} \right] \mathbf{W} \right) \\
&= \text{trace}\left(\mathbf{W}^\intercal \mathbf{S}_W \mathbf{W} \right)
\end{aligned}
$$

- Hàm mục tiêu trở thành:

$$J(\mathbf{W}) = \frac{\mathbf{S}_{\text{between}}}{\mathbf{S}_{\text{within}}} = \frac{\text{trace}(\mathbf{W}^\intercal \mathbf{S}_B \mathbf{W})}{\text{trace}\,(\mathbf{W}^\intercal \mathbf{S}_W \mathbf{W})}$$

- Áp dụng công thức đạo hàm của một phân thức và $\nabla_W \text{trace}(W^\intercal AW) = 2AW$ ta dễ dàng tính được:

$$\nabla_{\mathbf{W}} J(\mathbf{W}) = \frac{2\left(\mathbf{S}_B \mathbf{W} \text{trace}(\mathbf{W}^\intercal \mathbf{S}_W \mathbf{W}) - \text{trace}(\mathbf{W}^\intercal \mathbf{S}_B \mathbf{W}) \mathbf{S}_W \mathbf{W}\right)}{\left(\text{trace}(\mathbf{W}^\intercal \mathbf{S}_W \mathbf{W})\right)^2} = \mathbf{0}$$

$$\Leftrightarrow 2\mathbf{S}_B \mathbf{W} = 2 \underbrace{\frac{\text{trace}(\mathbf{W}^\intercal \mathbf{S}_B \mathbf{W})}{\text{trace}(\mathbf{W}^\intercal \mathbf{S}_W \mathbf{W})}}_{\lambda} \mathbf{S}_W \mathbf{W}$$

$$\Leftrightarrow \mathbf{S}_W^{-1} \mathbf{S}_B \mathbf{W} = \lambda \mathbf{W}$$

- Dòng thứ 2 suy ra thứ 3 là vì trace**$(\mathbf{W}^\intercal \mathbf{S}_B \mathbf{W})$trace / $(\mathbf{W}^\intercal \mathbf{S}_W \mathbf{W})$** chính là hàm mục tiêu J(w), hàm này có giá trị là một số vô hướng λ tại nghiệm tối ưu. Như vậy các cột của ma trận W chính là những véc tơ riêng tương ứng với trị riêng lớn nhất của ma trận $S^{-1}{}_W S_B$ và đồng thời những véc tơ cột này phải độc lập tuyến tính bởi nếu không thì phép chiếu lên không gian mới vẫn có thể biểu diễn qua một không gian có số chiều giảm nữa. Ngoài ra chúng ta còn chứng minh được rằng số chiều $K \leq C-1$. Tuy nhiên trong phạm vi hạn hẹp của cuốn sách chúng ta sẽ bỏ qua phần chứng minh này.

3. Dữ Liệu :

- Thêm một số thư viện vào để chạy thuật toán LDA:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sklearn
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
```

- Trong đó để train cho mô hình phân loại dựa vào thuật toán LDA trong sklearn thì cùng ta cần sử dụng class sklearn.discriminant_analysis.LinearDiscriminator.

```
LinearDiscriminantAnalysis(solver='svd',
shrinkage=None,
priors=None,
n_components=None,
store_covariance=False,
tol=0.0001,
covariance_estimator=None)
```

- Trong đó:
    - solver: Thuật toán được sử dụng để tìm thành phần chính. Bao gồm các lựa chọn svd (thuật toán phân tích suy biến), lsqr: dựa vào bình phương nhỏ nhất và eigen: dựa vào phép phân tích riêng (Eigenvalue decomposition).
    - shrinkage: tham số co. Tham số này được để là None nếu như covariance_estimator được sử dụng.
    - n_components: Số chiều trong không gian giảm chiều mà thuật toán sẽ giảm về.
    - tol: Ngưỡng tuyệt đối để một trị riêng của X được coi là quan trọng, được sử dụng để ước tính rank của X. Các chiều có trị riêng không quan trọng sẽ bị loại bỏ. Chỉ được sử dụng nếu solver là svd.
    - covariance_estimator: là dạng covariance_estimator được sử dụng để ước tính ma trận hiệp phương sai. Nếu None thì dựa vào công thức ước lượng hiệp phương sai thực nghiệm.
- Phương pháp LDA đối với bài toán phân loại đa lớp, chúng ta sẽ sử dụng thuật toán này nhằm phân lớp các loài hoa trên bộ dữ liệu iris dựa trên đầu vào là các thông số 'sepal-length', 'sepal-width', 'petal-length', 'petal-width'.
- Đọc dữ liệu dataset

```
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
cls = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'Class']
dataset = pd.read_csv(url, names=cls)
print(dataset)
```

	sepal-length	sepal-width	petal-length	petal-width	Class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
..	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

[150 rows x 5 columns]

- Tập dữ liệu bao gồm 150 hàng và 5 cột được lấy từ dataset iris dùng để phân lớp các loài hoa
- Tập dữ liệu được lấy từ nguồn https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data
- Phân chia tập dữ liệu thành các lớp và các biến mục tiêu:

```
X = dataset.iloc[:, 0:4].values
y = dataset.iloc[:, 4].values
```

- Xử lý tập dữ liệu phân chia thành train và test

```
sc = StandardScaler()
X = sc.fit_transform(X)
le = LabelEncoder()
y = le.fit_transform(y)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

- Áp Dụng LDA vào tập dữ liệu:

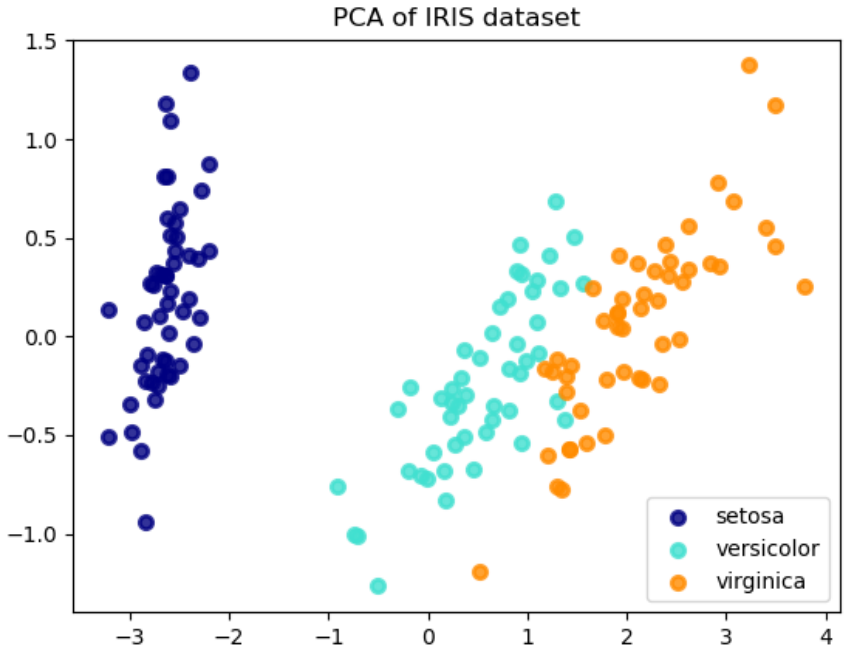
```
lda = LinearDiscriminantAnalysis(n_components=2)
X_train_reduce = lda.fit_transform(X_train, y_train)
X_test_reduce = lda.transform(X_test)
```

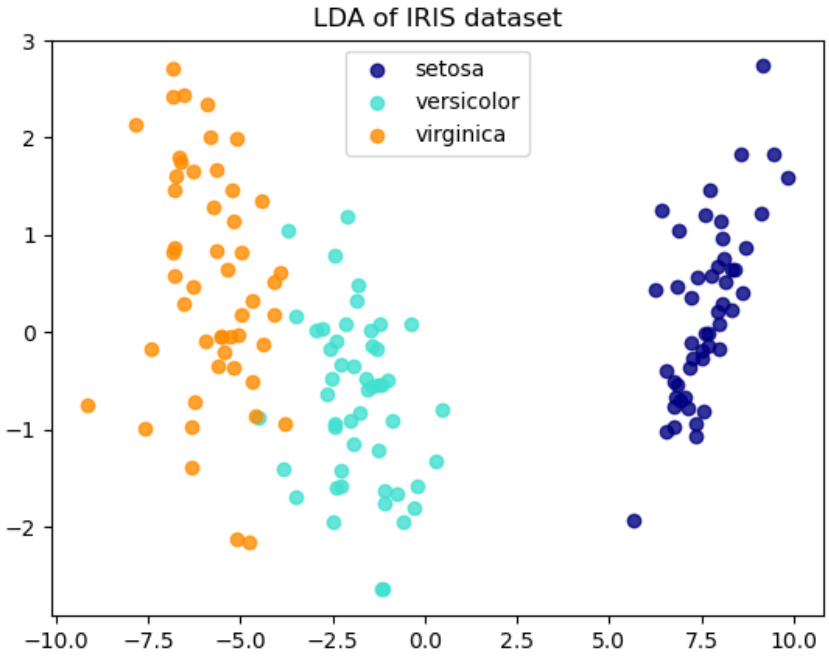
- Thực hiện vẽ biểu đồ và show ra kết quả biểu đồ:

```
35  plt.figure(figsize=(12, 8))
36  plt.scatter(
37      X_train_reduce[:,0],X_train_reduce[:,1],c=y_train,cmap='rainbow',
38    alpha=0.7,edgecolors='b'
39  )
40  plt.xlabel('dim 1')
41  plt.ylabel('dim 2')
42  plt.title('Two main dimensions of LDA on train dataset')
43  plt.show()
```

- Đa phần về cách tính toán được trình bày ở phần lý thuyết nên phần này các bạn thắc mắc cứ quay lại.

- So sánh PCA và LDA:





OUT:

```
explained variance ratio (first two components): [0.92461872 0.05306648]
```

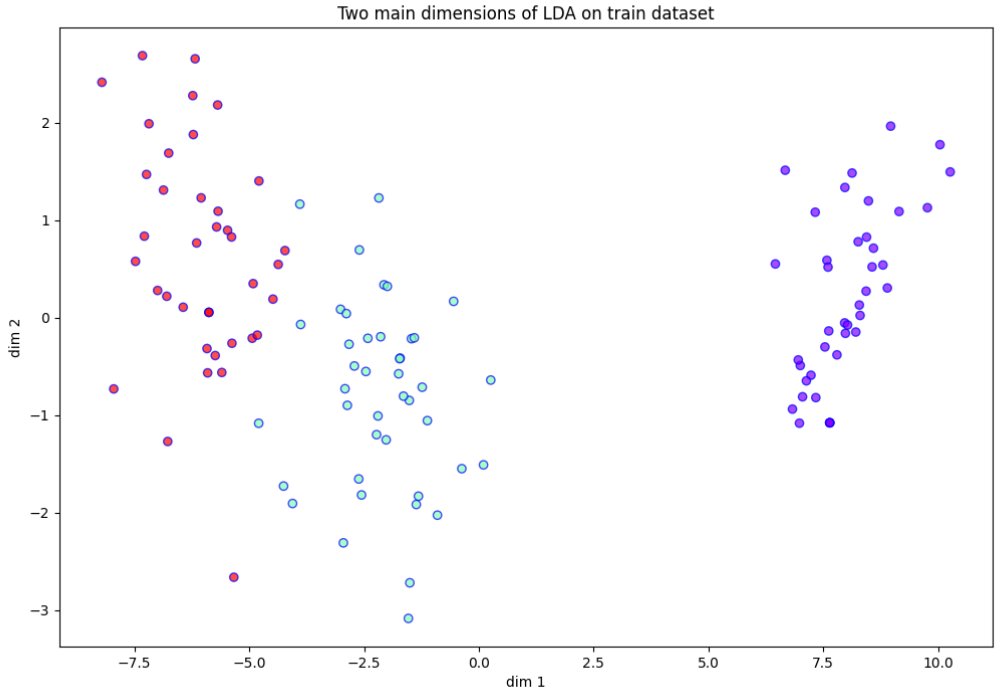
- Ở đây chúng ta thấy sự thay đổi trong sơ đồ giữa PCA và LDA:
    - **PCA:** được áp dụng cho dữ liệu này xác định sự kết hợp của các thuộc tính (thành phần chính hoặc hướng trong không gian đối

tượng địa lý) chiếm nhiều phương sai nhất trong dữ liệu. Ở đây chúng tôi vẽ biểu đồ các mẫu khác nhau trên 2 thành phần chính đầu tiên.

- **LDA:** cố gắng xác định các thuộc tính gây ra nhiều phương sai nhất giữa các lớp. Đặc biệt, LDA, trái ngược với PCA, là một phương pháp được giám sát, sử dụng các nhãn lớp đã biết.

4. Kết Quả:

- Sau khi sử dụng thuật toán LDA thì chúng ta thu được một biểu đồ biểu diễn phân loại các loài hoa trên tập dữ liệu iris và dựa trên các thông số đó là 'sepal-length', 'sepal-width', 'petal-length', 'petal-width'



- Để giải thuật toán LDA thì chúng ta dựa trên tối đa hóa hàm mục tiêu là trọng số giữa phương sai giữa các cụm và phương sai trên từng cụm. Tìm nghiệm tối ưu của hàm mục tiêu chính là đi tìm trị riêng lớn nhất của một ma trận vuông. Một điểm đáng chú ý đó là số chiều trong không gian giảm chiều tối đa có thể đạt được là $C-1$. Tính chất này cho thấy đối với bài toán có số lượng lớp càng ít thì số chiều có thể chiếu lên càng nhỏ. Chẳng hạn như kết quả của bài toán phân loại nhị phân này được chiếu lên không gian 1 chiều.

II. Kết Luận:

- LDA là một phương pháp giảm chiều dữ liệu nhưng kết hợp với thông tin về nhãn của biến mục tiêu. Mục đích của LDA đó là tìm ra những thành phần chính mà khi chiếu lên chúng, dữ liệu giữa các lớp là tách biệt nhất.

- Ý tưởng cơ bản của LDA là tìm một không gian mới với số chiều nhỏ hơn không gian ban đầu sao cho hình chiếu của các điểm trong cùng 1 class lên không gian mới này là gần nhau trong khi hình chiếu của các điểm của các classes khác nhau là khác nhau.
- Trong PCA, số chiều của không gian mới có thể là bất kỳ số nào không lớn hơn số chiều và số điểm của dữ liệu. Trong LDA, với bài toán có C classes, số chiều của không gian mới chỉ có thể không vượt quá C−1.
- LDA có giả sử ngầm rằng dữ liệu của các classes đều tuân theo phân phối chuẩn và các ma trận hiệp phương sai của các classes là gần nhau.
- Về Hạn chế thì LDA hoạt động rất tốt nếu các classes là linearly separable, tuy nhiên, chất lượng mô hình giảm đi rõ rệt nếu các classes là không linearly separable. Điều này dễ hiểu vì khi đó, chiếu dữ liệu lên phương nào thì cũng bị chồng lần, và việc tách biệt không thể thực hiện được như ở không gian ban đầu.
- Nhưng mặc dù cùng là thuật toán giảm chiều dữ liệu nhưng LDA lại phù hợp hơn PCA trong các bài toán học có giám sát, khi mà việc giảm chiều dữ liệu cần xét đến sự tách biệt giữa các lớp.

III. Tài Liệu Tham Khảo:

https://machinelearningcoban.com/2017/06/30/lda/#-vi-du-tren-python

https://en.wikipedia.org/wiki/Linear_discriminant_analysis

Code:
https://colab.research.google.com/drive/1Fh0IOFu90JXBMBI4PyOOTwUnauB4Eu7_