

**Báo cáo  
tiến độ  
lần 2  
Nhóm 13-TT**



# **ĐỀ TÀI : ĐIỂM DANH - CHẤM CÔNG BẰNG NHẬN DIỆN KHUÔNG MẶT**

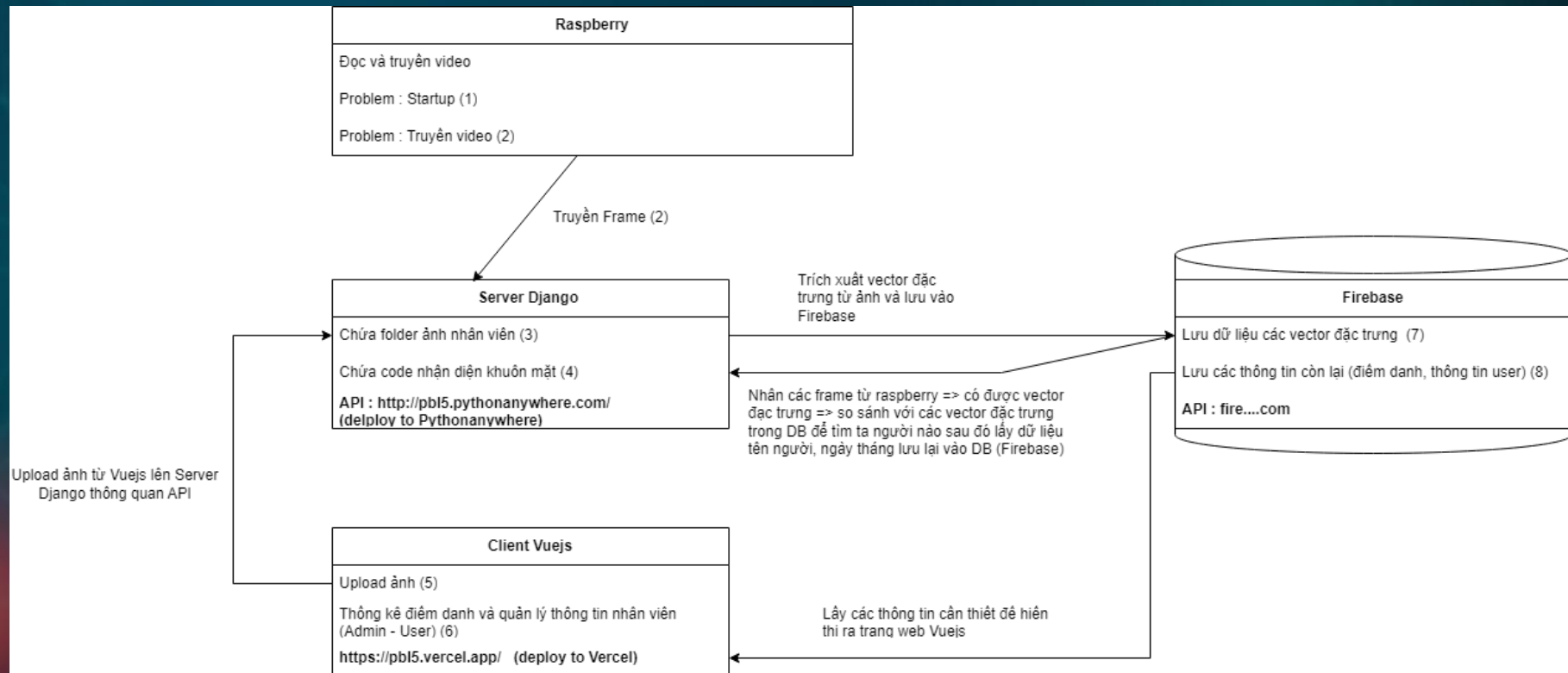
**Nội dung chính gồm các phần sau :**

**Phần I . Tổ chức dữ liệu trong firebase - Xây dựng client Vuejs (Nguyễn Văn Mạnh – Captain)**

**Phần II. Xây dựng REST API by Django (Nguyễn Công Cường)**

**Phần III. Xây dựng Model nhận diện khuôn mặt (Nguyễn Văn Hoàng Phúc – Trần Thanh Nguyên)**

# Nhất lại Sơ đồ hệ thống



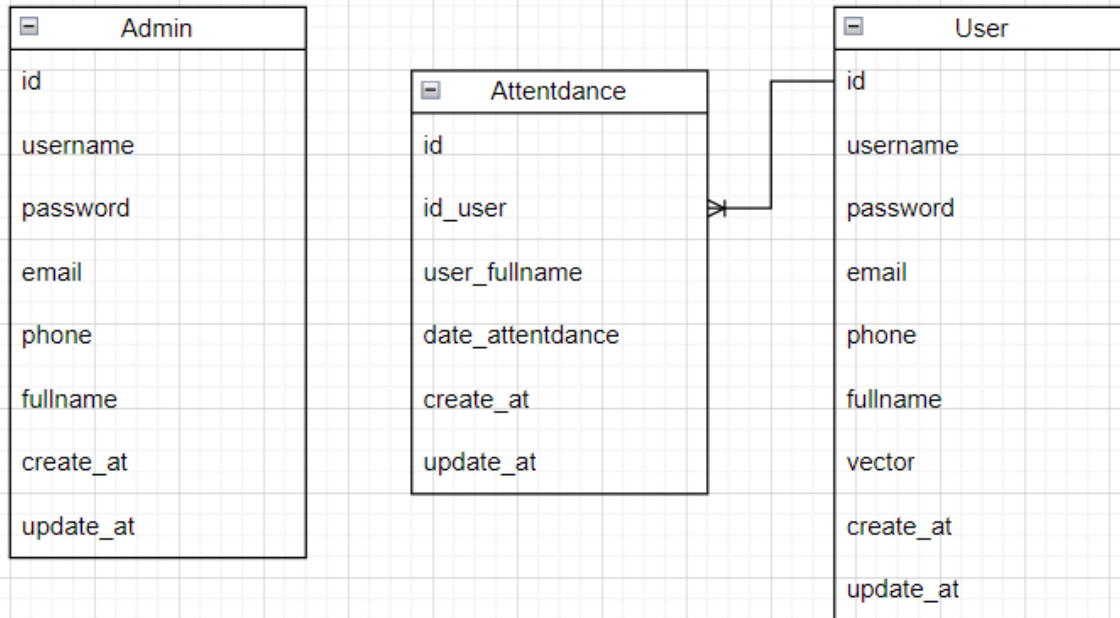
# Tổ chức dữ liệu trong firebase & Xây dựng client Vuejs

# 01


---



Nguyễn Văn Mạnh – Captain

# Firestore - Diagram





# Firebase


 **Firebase**


 **Project Overview** 

Project shortcuts


 **Authentication**


 **Firestore Database**


 **Hosting**


 **Realtime Database**


Product categories




**Build** 

**Release & Monitor** 



**Analytics** 







**Engage** 

 **All products**

PBL5  Cloud Firestore  

**Panel view** Query builder

 > admins > 3z3Az1bCaRgA0. [More in Google Cloud](#) 

 pbl5-35471	 admins  	 3z3Az1bCaRgA0Als4h5 
<a href="#">+ Start collection</a>	<a href="#">+ Add document</a>	<a href="#">+ Start collection</a>
admins >	3z3Az1bCaRgA0Als4h5 >	<a href="#">+ Add field</a>
attendance	BHJp1dmeFo0tPqG7rT86	create_at: March 14, 2023 at 9:48:22 AM UTC+7
users	Pa2NTvciYQJOWhosdPm6	email: "vanmanhadmin@gmail.com"
	Uxqf9xf4i1r8dtNujRJq	fullname: "Nguyễn Văn Mạnh Admin"
	Z5IJ0pzq1FIEPCpvYzkU	password: "\$2a\$10\$Nf/CKKH8c3a/l/va7BWAeK1UyfD72GmPaqIHrc3ljdJZhwUxU"
	cMI1z8TyvznTpokY1qxN	phone: "01236596245"
	cttocsCMJFVVVRVD7K07q	update_at: March 14, 2023 at 9:48:46 AM UTC+7

# Admins

P

Firestore

Synced • in a few seconds >

⋮

Preview

Share

Search collections, actions...

<<

Filter

Sort

Hide fields

API

Search data...

+ New Record

COLLECTIONS

Admins

Attendances

Users

SQL COLLECTIONS ⓘ

+ Create with SQL query

DOCUMENT PATH	DOCUMENT ID	CREATE AT	EMAIL	FULLNAME	PASSWORD	PHONE	UPDATE AT	CREATED TIME	UPDATED TIME	NEW COLUMN
projects/pbl5-35471/datab...	cttocsCMjFVVRVD7K07q	03/14/2023 01:11	adminconguong@gmail.c...	---	\$2a\$10\$SvWr3vM45rvgxjvNT...	---	03/14/2023 01:11	03/14/2023 01:11	03/14/2023 01:11	
projects/pbl5-35471/datab...	cMI1z8TyvznTpokY1qxN	03/14/2023 01:10	admin3@gmail.com	---	\$2a\$10\$hPzDf3tgjs/c6WG7S...	---	03/14/2023 01:10	03/14/2023 01:10	03/14/2023 01:10	
projects/pbl5-35471/datab...	Z5IJ0pzq1FIEPCpvYzkU	03/14/2023 01:11	adminthanhnguyen@gmai...	---	\$2a\$10\$anzxA0F8srBlzkyvF...	---	03/14/2023 01:11	03/14/2023 01:11	03/14/2023 01:11	
projects/pbl5-35471/datab...	Uxqf9xf4i1r8dtNujRJq	03/14/2023 01:11	adminhoangphuc@gmail.c...	---	\$2a\$10\$aLpyc/GyWGZFfrS8...	---	03/14/2023 01:11	03/14/2023 01:11	03/14/2023 01:11	
projects/pbl5-35471/datab...	Pa2NTvciYQJOWhosdPm6	03/14/2023 01:10	admin1@gmail.com	Admin Nguyễn Văn Mạnh	\$2a\$10\$AR/edpzKt20bLQjo...	0971404372	03/14/2023 09:43	03/14/2023 01:10	03/14/2023 09:43	
projects/pbl5-35471/datab...	BHJlp1dmeFoOtPqG7rT86	03/14/2023 01:10	adminvanmanh@gmail.com	---	\$2a\$10\$DMcb.5D.D8okwHj5...	---	03/14/2023 01:10	03/14/2023 01:10	03/14/2023 01:10	
projects/pbl5-35471/datab...	3z3AzlbCaRgA0Als4h5	03/14/2023 09:48	vanmanhadmin@gmail.com	Nguyễn Văn Mạnh Admin	\$2a\$10\$nF/CKKH8c3a/l./va...	01236596245	03/14/2023 09:48	03/14/2023 09:48	03/14/2023 09:48	
+ NEW RECORD										

# Users



# Attendances

Firestore

Synced • in a few seconds >

Preview

Share

Search collections, actions...

FilterSortHide fields

API

Search data...

+ New Record

COLLECTIONS

Admins

Attendances

Users

SQL COLLECTIONS ?

+ Create with SQL query

DOCUMENT PATH	DOCUMENT ID	CREATE AT	DATE ATTENDANCE	ID USER	UPDATE AT	USER FULLNAME	CREATED TIME	UPDATED TIME	NEW COLUMN
projects/pbl5-35471/databases...	xsH6WzII7TaHfbQZ1zxb	2/3/2023	2/3/2023	Zo4rZTgOyGx1PEqDEpyb	2/3/2023	Nguyễn Văn Mạnh	03/01/2023 08:21	03/01/2023 08:22	
projects/pbl5-35471/databases...	U6u6KKneohFmORgE1DUc	2/3/2023	2/3/2023	jxyufm5FWN6bHxinObIk	2/3/2023	Nguyễn Văn Hoàng Phúc	03/01/2023 08:22	03/01/2023 08:24	
+ NEW RECORD									



# Client Vuejs



**Demo User : <https://pbl-5.vercel.app/>**

**Email : user123@gmail.com**

**Password : user123**

**Demo Admin : <https://pbl-5.vercel.app/admin>**

**Email : admin1@gmail.com**

**Password : admin1**

# Xây dựng REST API by Django

# 02

---

Nguyễn Công Cường

# 1. Ứng dụng của Django vào back-end web nhận diện khuôn mặt

- ❖ Quản lý cơ sở dữ liệu: Django cung cấp một ORM (Object-Relational Mapping) để truy cập cơ sở dữ liệu. Điều này giúp cho việc lưu trữ và truy xuất dữ liệu trở nên dễ dàng hơn.
- ❖ Ở đây chúng em sử dụng Django kết hợp với Firebase để lưu trữ dữ liệu. Một trong những thư viện phổ biến để tương tác với Firebase trong Django là "python-firebase".

# 1. Ứng dụng của Django vào back-end web nhận diện khuôn mặt

- ⬡ Xử lý yêu cầu HTTP: Django cung cấp một bộ xử lý yêu cầu HTTP mạnh mẽ để xử lý các yêu cầu đến từ các client. Điều này giúp cho việc xử lý các yêu cầu nhận diện khuôn mặt từ các client được thực hiện một cách hiệu quả.
- ⬡ Phát triển ứng dụng web API: Django cho phép phát triển ứng dụng web API để truy cập dữ liệu từ các client. Điều này giúp cho việc xử lý các yêu cầu nhận diện khuôn mặt và trả về kết quả dễ dàng hơn.

# 1. Ứng dụng của Django vào back-end web nhận diện khuôn mặt

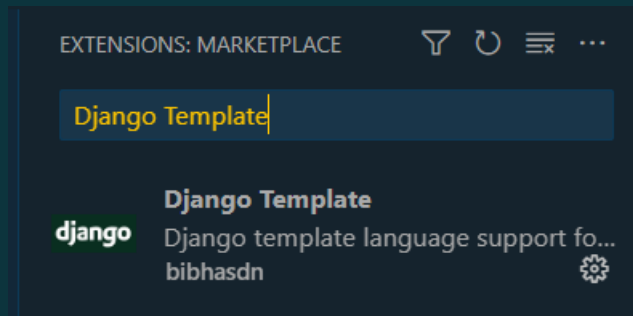
- ⬡ Tính bảo mật cao: Django có nhiều tính năng bảo mật để giữ cho ứng dụng an toàn. Điều này đặc biệt quan trọng khi đang xây dựng một ứng dụng nhận diện khuôn mặt, vì thông tin về khuôn mặt của các nhân viên là nhạy cảm và cần được bảo vệ.
- ⬡ Như vậy ta có thể thấy được Django là một framework phù hợp để phát triển back-end web nhận diện khuôn mặt dùng cho chấm công trong công ty.

## 2. Các bước đã tiến hành với Django

### ⬡ Cài đặt thư viện Django trong Python

```
PS E:\Django> pip install Django
Requirement already satisfied: Django in c:\users\win10hdd\appdata\local\programs\python\python311\li
Requirement already satisfied: asgiref<4,>=3.5.2 in c:\users\win10hdd\appdata\local\programs\python\p
(3.6.0)
Requirement already satisfied: sqlparse>=0.2.2 in c:\users\win10hdd\appdata\local\programs\python\pyt
0.4.3)
Requirement already satisfied: tzdata in c:\users\win10hdd\appdata\local\programs\python\python311\li
```

### ⬡ Cài extension Django template



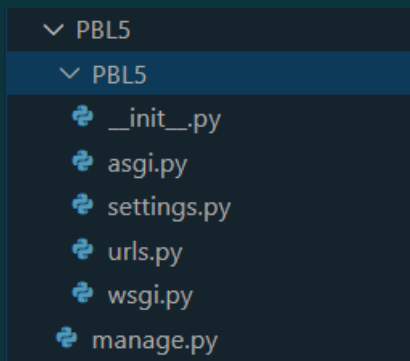


## 2. Các bước đã tiến hành với Django

### ⬡ Tạo project PBL5

```
[notice] A new release of pip is available: 23.0 -> 23.0.1  
[notice] To update, run: python.exe -m pip install --upgrade pip  
PS E:\Django> django-admin startproject PBL5  
PS E:\Django> 
```

### ⬡ Cấu trúc cây ban đầu của project PBL5



## 2. Các bước đã tiến hành với Django

### ⬡ Chạy server Django PBL5

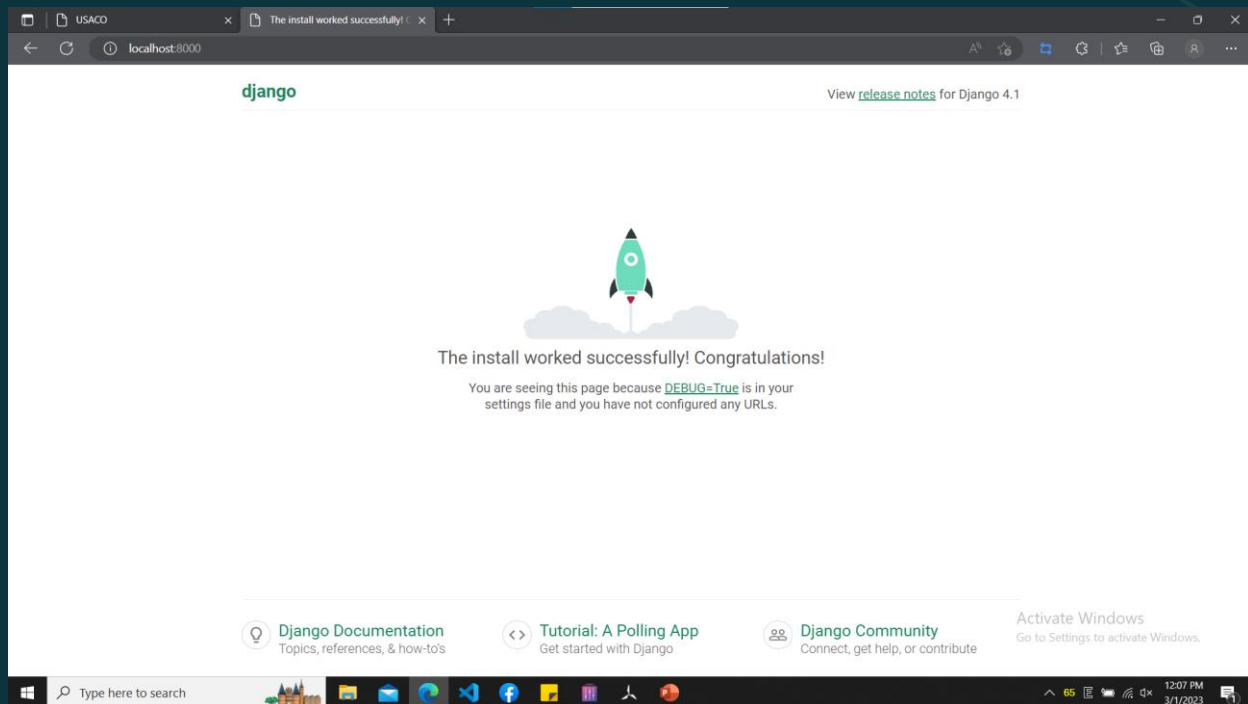
```
PS E:\Django> cd PBL5
PS E:\Django\PBL5> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply
sessions.
Run 'python manage.py migrate' to apply them.
March 01, 2023 - 11:11:37
Django version 4.1.7, using settings 'PBL5.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

## 2. Các bước đã tiến hành với Django

⬡ Server tại `http://127.0.0.1:8000/`



## 5. Tài liệu tham khảo:

- ⬡ Chat GPT
- ⬡ [Django documentation | Django documentation | Django \(djangoproject.com\)](#)
- ⬡ [Django Tutorial \(w3schools.com\)](#)

# Xây dựng Model nhận diện khuôn mặt

# 03

---

Nguyễn Văn Hoàng Phúc – Trần Thanh Nguyên

# Phần thuật toán:

Đầu vào:

Tập dữ liệu huấn luyện (các khuôn mặt được chọn làm gốc để nhận diện)

Tập dữ liệu kiểm thử (các khuôn mặt được qua từ camera cần nhận diện)

Đầu ra:

Kết quả nhận diện trên tập dữ liệu kiểm thử  
Hiệu suất nhận diện bao nhiêu phần trăm

## Phần thuật toán:

1. Trên tập dữ liệu huấn luyện thực hiện trích xuất đặc trưng các khuôn mặt.
  - + Sử dụng thư viện `face_recognition` để trích xuất đặc trưng của khuôn mặt là một vector 128 phần tử lưu vào file `know_face_encode.csv`
2. Trên tập dữ liệu kiểm thử thực hiện trích xuất đặc trưng các khuôn mặt.
  - + Sử dụng thư viện `face_recognition` để trích xuất đặc trưng của khuôn mặt là một vector 128 phần tử lưu vào file `not_know_face_encode.csv`
  - + Sau đó sử dụng các mô hình ML (GaussianNB, KNN, PCA, LDA, ...) để nhận diện các đặc trưng và thêm 1 trường vào cuối là trường nhận diện được (tên hoặc id của người)
3. So sánh 2 trường cuối cùng của file `not_know_face_encode.csv` để tính ra hiệu suất của từng mô hình ML.
4. Chọn các mô hình có hiệu suất tốt nhất và kết hợp để tăng hiệu suất.

# Sử dụng thư viện face\_recognition

1. Thực hiện trích xuất đặc trưng các khuôn mặt.

```
encode = face_recognition.face_encodings(img)[0]
```

2. Nhận diện đặt trưng của khuôn mặt chưa có từ danh sách các đặc trưng của các khuôn mặt đã có

```
matches = face_recognition.compare_faces(encodeListKnow, encodeFaceNotKnow)
```

3. Tính khoảng cách để xác định phần trăm độ chính xác với các đặc trưng đã có.

```
faceDis = face_recognition.face_distance(encodeListKnow, encodeFaceNotKnow)
```



# Sử dụng thư viện face\_recognition

```
def face_distance (known_encodings, face_encoding):  
    """  
    Tính toán khoảng cách giữa một khuôn mặt mới và các khuôn mặt đã biết.  
    :param known_encodings: danh sách các vector mã hóa khuôn mặt đã biết.  
    :param face_encoding: vector mã hóa khuôn mặt của khuôn mặt mới.  
    :return: danh sách các giá trị khoảng cách giữa khuôn mặt mới và các khuôn mặt đã biết.  
    """  
  
    # Chuyển đổi danh sách mã hóa khuôn mặt đã biết và mã hóa khuôn mặt mới sang các mảng NumPy.  
    known_encodings = np.array(known_encodings)  
    face_encoding = np.array(face_encoding)  
  
    # Tính toán khoảng cách Euclidean giữa khuôn mặt mới và các khuôn mặt đã biết.  
    distances = np.linalg.norm(known_encodings - face_encoding, axis=1)  
  
    # Trả về danh sách khoảng cách giữa khuôn mặt mới và các khuôn mặt đã biết.  
    return distances
```

Việc sử dụng hàm face\_distance của thư viện face\_recognition để nhận diện khuôn mặt chưa chắc đã đạt hiệu suất tốt nhất nên cần cải thiện bằng các mô hình ML khác nhau.

# Phát triển thuật toán

- Thực hiện giảm chiều dữ liệu - PP PCA
- Sử dụng các mô hình Machine Learning cải thiện hiệu suất:

01

**PCA**

Principal Component Analysis

03

**LDA**

Linear Discriminant Analysis

02

**KNN**

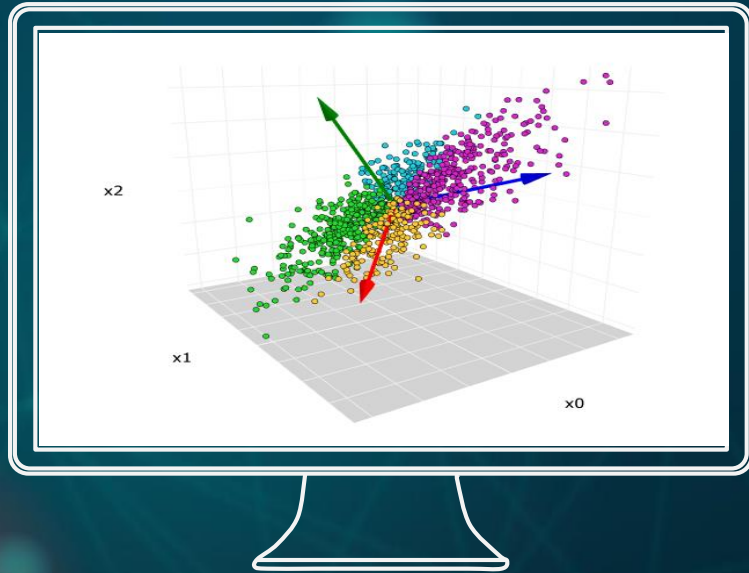
k-Nearest Neighbor

04

**MLD, SGD, SVM**

Các loại mô hình ML khác

# Giảm chiều dữ liệu - Phương pháp PCA



PCA( Principal Component Analysis) : là phương pháp chọn ra những đặt trưng quan trọng

Mã trận đặt trưng hiện tại của các ảnh là mã trận 1 x 128 (vẫn còn khá lớn)

# Các bước của PCA



Hiện tại có nhiều thư viện hỗ trợ PCA : như numpy , scikit-learn pandas...

# KNN (k-Nearest Neighbor)

**Định nghĩa :** là một thuật toán học có giám sát dùng để phân loại hoặc dự đoán dữ liệu mới dựa trên các điểm dữ liệu đã biết.

**Sử dụng phương pháp :** tính khoảng cách , dựa vào nhãn các k điểm gần nhất để đưa ra dự đoán nhãn của điểm này

**Thư viện sciki-learn:** có hàm `kneighborsclassified` hỗ trợ tính toán

# LDA (Linear Discriminant Analysis)

```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
import numpy as np

# Các vector đặc trưng của các khuôn mặt đã biết
known_faces = np.array(encodeListKnow)

# Nhãn của các khuôn mặt đã biết
known_labels = np.array(classnames)

# Các vector đặc trưng của các khuôn mặt chưa biết
unknown_faces = np.array(encodeListUnknow)

# Khởi tạo mô hình LDA
lda = LinearDiscriminantAnalysis()

# Huấn luyện mô hình LDA với các vector đặc trưng đã biết và nhãn tương ứng
lda.fit(known_faces, known_labels)

# Dự đoán nhãn của các khuôn mặt chưa biết bằng cách sử dụng mô hình LDA đã huấn luyện
predicted_labels = lda.predict(unknown_faces)

# In ra các nhãn được dự đoán
print(predicted_labels)
```

# LDA (Linear Discriminant Analysis)

Kết quả:

```
C:\ProgramData\Anaconda3\envs\AnacondaFace\python.exe C:\Users\ACER\PycharmProjects\AnacondaFace\LDA.py
['Charlie_Puth', 'Charlie_Puth', 'Charlie_Puth', 'Elon_Musk', 'Elon_Musk', 'HPhuc', 'HPhuc', 'Taylor_Swift', 'Taylor_Swift', 'Taylor_Swift']
guess: ['Charlie_Puth' 'Charlie_Puth' 'Charlie_Puth' 'Elon_Musk' 'Elon_Musk'
        'HPhuc' 'Taylor_Swift']
correct: ['Charlie_Puth' 'Charlie_Puth' 'Charlie_Puth' 'Elon_Musk' 'Elon_Musk'
          'HPhuc' 'Taylor_Swift']

Process finished with exit code 0
```

Nhận xét:

- Hiệu suất là 100% nhận được nhận dạng chính xác.
- Vì dữ liệu còn nhỏ nên độ chính xác cao. Cần thêm dữ liệu để đánh giá hiệu suất tốt hơn.

# SVM (Support Vector Machine)

**Định nghĩa :** là một thuật toán học có giám sát được sử dụng cho các bài toán phân loại và hồi quy. Cơ chế hoạt động của SVM được thực hiện bằng cách tìm một siêu mặt phẳng (hyperplane) phân chia tốt nhất các điểm dữ liệu thuộc các lớp khác nhau

## Các bước tiến hành :

Lựa chọn kernel function và các siêu tham số (hyperparameters) cho SVM.

Huấn luyện SVM trên dữ liệu huấn luyện.

Đánh giá hiệu suất của SVM trên dữ liệu kiểm tra.

Tinh chỉnh các siêu tham số của SVM để cải thiện hiệu suất.

**Thư viện scikit-learn:** có hàm SVC hỗ trợ tính toán

