

Nhập môn học máy và khai phá dữ liệu

Dự đoán nhạc cụ dựa trên âm thanh



[Date]

[Company name]

[Company address]

**Báo cáo môn học  
Nhập môn học máy và khai phá dữ liệu**

Đề tài: Nhận diện nhạc cụ bằng âm thanh

Nguyễn Đức Anh – 20172937  
Lê Tùng Lâm – 20173218  
Phùng Ngọc Minh – 20173257

# Tóm tắt

Trong báo cáo này nhóm chúng em sẽ trình bày cách sử dụng một vài mô hình học máy để áp dụng vào việc nhận diện nhạc cụ dựa trên âm thanh của chúng. Việc này được thực hiện bằng cách trích chọn ra các đặc trưng của các nhạc cụ, chẳng hạn như Spectrum, MFCC, Chroma, Contrast,...,sau đó sử dụng các mô hình học máy để thực hiện việc phân loại và dự đoán. Trong phạm vi đề tài này, chúng em xây dựng một mô hình có thể nhận diện được nhạc cụ với dữ liệu là các tệp tin âm thanh ngắn có độ dài 4 giây. Các thuộc tính được trích xuất từ các tệp tin này sau đó được tiền xử lý trước khi đưa vào các mô hình: KNN, SVM, Random Forest. Cuối cùng, đánh giá để chọn ra mô hình tốt nhất.

# Mô tả bài toán

Hiện nay, việc nghe nhạc qua băng đĩa, đài cassette, radio truyền thống đã dần được thay thế bằng các dịch vụ nghe nhạc trực tuyến. Các nền tảng phổ biến như Youtube, Apple Music, Spotify đang chiếm thị phần lớn. Sự khác biệt làm nên thành công của các nền tảng này là hệ thống gợi ý nhạc, trong đó gợi ý theo nhạc cụ là một mảng quan trọng. Để gợi ý theo nhạc cụ, cần có một hệ thống nhận diện và phân loại nhạc cụ hiệu quả. Do đó, nhóm lựa chọn tìm hiểu về đề tài **“Nhận diện nhạc cụ qua dữ liệu âm thanh”**.

# Dữ liệu

Bộ dữ liệu được sử dụng là Nsynth của Google. Bộ dữ liệu này gồm 305979 nốt nhạc, mỗi nốt nhạc có cao độ, âm sắc riêng và được lưu trữ độc lập ở tệp tin âm thanh định dạng ".wav" dài 4 giây. Tất cả dữ liệu thuộc 11 loại nhạc cụ khác nhau: bass, brass, flute, guitar, keyboard, mallet, organ, reed, string, synth và vocal. Sau khi tải về và xử lý, bộ dữ liệu đã được chia thành các tập train, validate và test với tỉ lệ lần lượt là 247603 tệp (93.7%), 12678 tệp (4.8%) và 4096 tệp (1.5%). Nsynth có thể được tải xuống ở địa chỉ sau: "https://magenta.tensorflow.org/datasets/nsynth".

Chart, bar chart

Description automatically generated

Hình 1. Phân phối dữ liệu

# Các đặc trưng của âm thanh

Chúng ta định nghĩa âm thanh là những dao động cơ học lan truyền trong không khí, chất lỏng,  
chất rắn và các môi trường truyền âm. Mỗi âm thanh có những đặc trưng riêng như tần số, âm  
sắc,... Vì vậy, để có thể phân biệt được các loại nhạc cụ dựa vào âm thanh thì chúng ta cần phải  
có hiểu biết về các đặc trưng riêng của âm thanh. Ở đây chúng ta thao tác với tệp tin âm thanh  
(.wav) và sử dụng thư viện Librosa (Python) xử lý các tệp ".wav" để trích ra các đặc trưng.

## **Harmonic** Âm harmonic là âm có tần số là bội số nguyên của âm cơ bản. Ví dụ âm cơ bản có tần số 50Hz thì các âm harmonic có thể là 100Hz, 150Hz, 200Hz,...Một nhạc cụ có tính harmonic thì ta sẽ thấy âm phát ra của chúng rất vang. Một nhạc cụ gõ (percussive) thì có thể là bất kể thứ gì mà ta có thể gõ được. Chúng ta có thể tách tệp sóng (.wav) ra thành hai đặc trưng harmonic và percussive. Sử dụng Librosa ta có các hình minh họa như dưới đây.

Chart

Description automatically generated

Hình 2. Biên độ harmonic và percusive của mallet

Chart

Description automatically generated

Hình 3. Biên độ harmonic và percusive của guitar

Chart, line chart

Description automatically generated

Hình 4. Biên độ harmonic và percusive của keyboard

A picture containing chart

Description automatically generated

Hình 5. Biên độ harmonic và percusive của organ

## **Zero Crossing Rate**

Số lần tín hiệu âm thanh đi qua mốc 0.

## **MFCC**

Trong xử lý âm thanh, mel-frequency cepstrum (MFC) là sự thể hiện phổ công suất ngắn hạn của âm thanh, dựa trên sự biến đổi cosin tuyến tính của phổ công suất log trên thang tần số mel phi tuyến tính. Mel-frequency cepstral coefficients (MFCC) là các hệ số tạo nên MFC.

## **Chroma**

Độ cao của một âm được đặc trưng bởi tốc độ rung của âm thanh đó. Hai âm được gọi là cùng sắc độ nếu như cao độ của chúng cách nhau một quãng tám (octave). Dựa trên quan sát này, cao độ có thể được tách ra làm hai phần, độ cao của tone và sắc độ (chroma).

## **Spectrogram**

Spectrogram là một công cụ cơ bản trong phân tích phổ âm thanh và các lĩnh vực khác. Nó đã được ứng dụng trong phân tích giọng nói. Spectrogram có thể được định nghĩa như một biểu đồ cường độ của Short Time Fourier Transform (STFT).

## **Contrast**

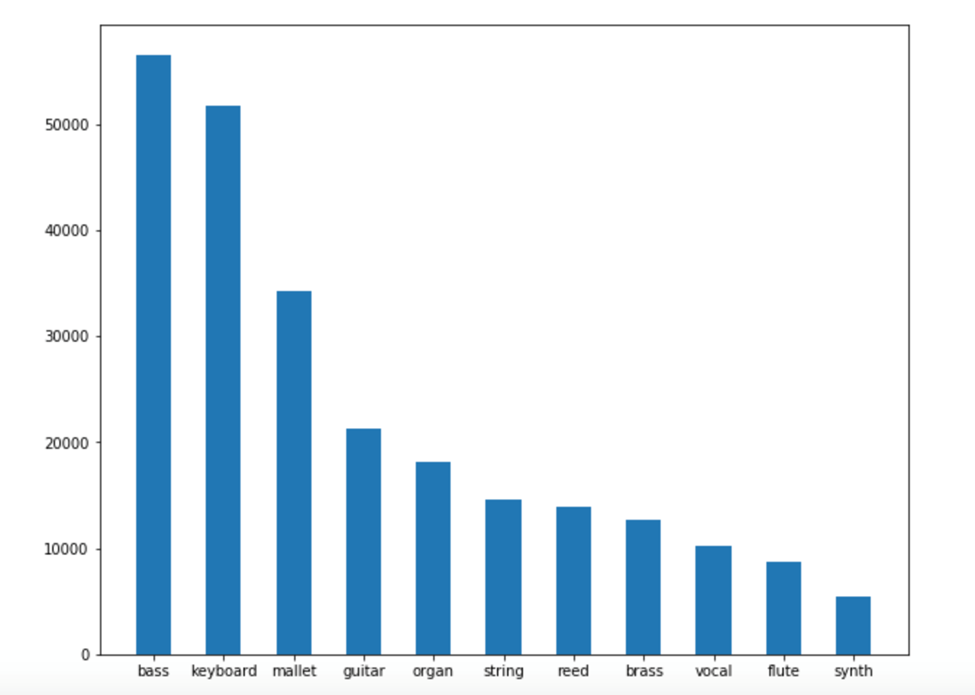
Một đặc trưng khác nữa là contrast, chỉ ra sự tương phản về phổ âm thanh. Ở đây nhóm chủ yếu sử dụng thư viện Librosa để tách ra các đặc trưng của âm thanh nên sẽ không đi sâu chi tiết các đặc trưng đó là gì.

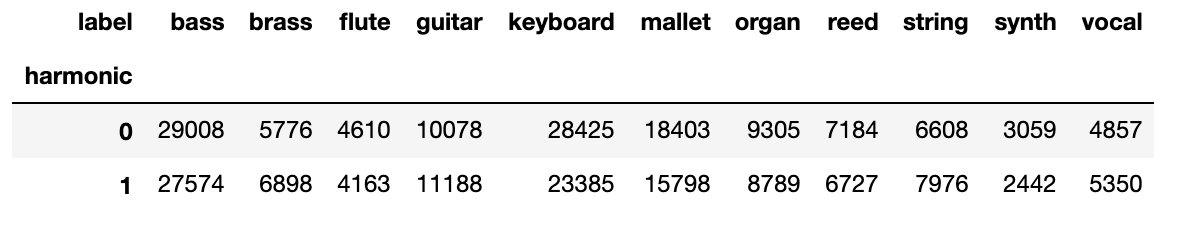
# **Trích chọn các đặc trưng từ tệp tin âm thanh**

Librosa là một gói Python dành cho việc phân tích âm thanh. Gói này cung cấp các khối cần thiết cho việc tạo ra các hệ thống truy xuất thông tin âm nhạc. Librosa có thể được tìm thấy ở trang chủ [https://librosa.org/doc/latest/index.html#](https://librosa.org/doc/latest/index.html).

Sử dụng gói thư viện Librosa, nhóm đã tách ra được các đặc trưng của tệp tin âm thanh và đưa về dạng sau

# **Thống kê và trực quan hóa dữ liệu**





# **Tiền xử lý dữ liệu**

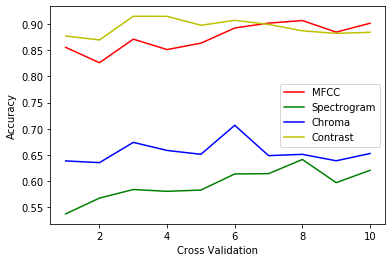
* 1. Mã hoá nhãn:   
     Các nhãn là tên các loại nhạc cụ được mã hoá thành các số nguyên từ 0-10.
  2. Chuẩn hoá:

Các thuộc tính được sử dụng sẽ được chuẩn hoá để có trung bình bằng 0 và độ lệch chuẩn bằng 1

# **Lựa chọn thuộc tính**

Sử dụng một mô hình đơn giản KNN với k = 1 trên tập test với chiến lược 10-fold cross-validate để lựa chọn giữa các thuộc tính dạng mảng.

Kết quả như sau:



* MFCC và Contrast tốt hơn so với 2 thuộc tính còn lại, về cả độ chính xác và độ phức tạp.
* Độ Harmonic trung bình của các loại nhạc cụ không chênh lệch nhiều so với độ percussive nên không lựa chọn thuộc tính này
* Zero-Crossing rate được sử dụng

Sau khi lựa chọn được các thuộc tính sử dụng, nhóm em chuẩn hoá từng thuộc tính và kết hợp lại để được features vector có kích thước 21

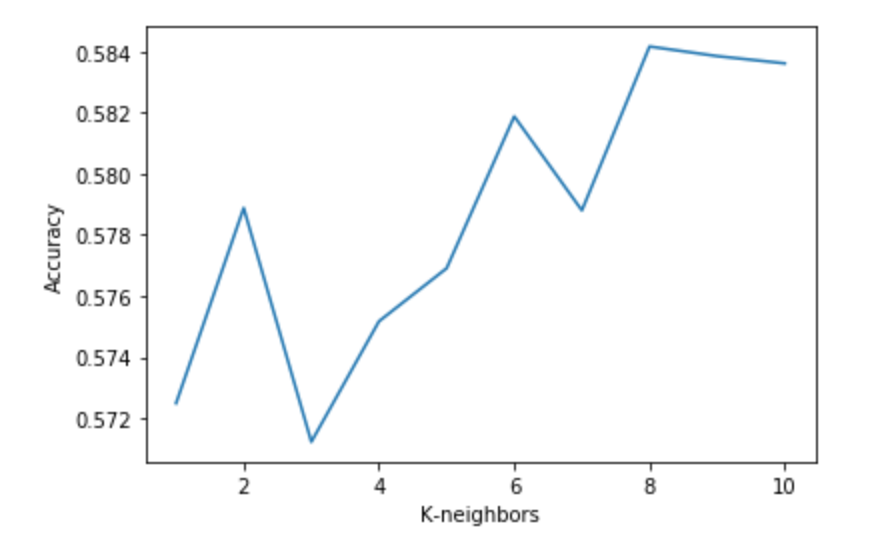
# Lựa chọn và huấn luyện các mô hình và đánh giá

* 1. KNN

Với KNN nhóm em thực hiện quy trình sau: huấn luyện mô hình trên dữ liệu train, lựa chọn ra bộ siêu tham số tốt nhất đối với mô hình đó trên tập validate, huấn luyện lại mô hình trên tập train+ tập validate và sử dụng tập test để chọn ra mô hình tốt nhất.

Siêu tham số cần học là số lượng neighbors k.

Kết quả sau khi huấn luyện trên tập train và đánh giá trên tập validate:



Với k = 8, mô hình cho kết quả tốt nhất. Sử dụng mô hình KNN với k = 8 train lại trên tập train+validate và dự đoán với tập test, được độ chính xác 89%.

Confusion matrix:

Scatter chart

Description automatically generated

* 1. SVM và Random forest

Đối với các mô hình phức tạp như SVM và Random Forest, do lượng dữ liệu quá lớn nên không đủ thời gian huấn luyện trên toàn bộ tập train. Vì vậy, nhóm em huấn luyện trên tập validate và đánh giá trên tập test

Các tham số học được với SVM: C = 1,000,000; gamma = 0.01

Độ chính xác khi đánh giá mô hình SVM này trên tập test là 99% với 4045 dữ liệu dự đoán đúng

Confusion matrix với SVM:

Chart

Description automatically generated

Chart

Description automatically generated

Các tham số học được với Random Forest:

n\_estimators=100, min\_samples\_leaf=1, min\_samples\_split=5, max\_depth=20

Độ chính xác khi đánh giá trên tập test là 99% với 4063 dữ liệu dự đoán đúng

# Kết luận

Trong đề tài này, nhóm chúng em đã sử dụng các mô hình dự đoán KNN, SVM, Random forest. Trong khi huấn luyện thì KNN có thể chạy hết toàn bộ dữ liệu và cho kết quả tốt, còn SVM và Random Forest thì do mô hình phức tạp và bộ dữ liệu quá lớn nên chỉ có thể huấn luyện trên tập validate và cũng cho kết quả tốt trên tập test.

# Phân công nhiệm vụ

Nguyễn Đức Anh: Crawl dữ liệu, trích xuất và lưu trữ các feature để chuẩn bị đưa vào học

Phùng Ngọc Minh: Xây dựng mô hình học máy, đánh giá

Lê Tùng Lâm: Viết báo cáo