TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Đề tài: Ứng dụng học máy trong xây dựng dữ liệu với xác thực tiếng nói

Học phần: Nhập môn Học máy và khai phá dữ liệu

Giảng viên: **PGS.TS Thân Quang Khoát**

Nhóm sinh viên thực hiện:

STT	Họ và tên	MSSV
1	Hữu Tường Tú	20194395
2	Nguyễn Phúc Tân	20194163
3	Nguyễn Thành Phong	20192016

Hà Nội, tháng 7 năm 2022

Tổng quan

Xác thực tiếng nói : Đây là task trong xử lý ngôn ngữ tự nhiên đầu vào là giọng nói của 1 người và danh tính của một người. Cần phải xác nhận là giọng nói và danh tính này có phải cùng là một người hay không. Việc xây dựng và thu thập dữ liệu sẽ chiếm một phần quan trọng trong task xác thực tiếng nói này. Với dữ liệu tốt ta có thể xây dựng ra một mô hình tốt từ đó việc xác thực tiếng nói sẽ trở nên chính xác hơn. Do đó chúng em đề xuất đề tài "Ứng dụng học máy trong xây dựng dữ liệu đối với xác thực tiếng nói" nhằm xây dựng một bộ dữ liệu cho task này.

Source code và hướng dẫn chạy được lưu tại: huutuongtu/ML_project (github.com)

Giới thiệu

Bài viết được chia làm 4 phần:

- Phần một: Giới thiệu về xác thực tiếng nói (Speaker Verification).
- Phần hai: Giới thiệu về hai pre-trained được dùng trong project (VAD, X-vector).
- Phần ba: Pipeline project, các thuật toán được sử dụng để đánh nhãn dữ liệu, các bước xử lý dữ liệu và huấn luyện dữ liệu cùng kết quả thu được.
- Phần cuối: Tổng kết. Phần cuối sẽ nêu ra những gì tổng hợp được trong quá trình nghiên cứu.

MỤC LỤC

CHƯƠNG 1

GIỚI THIỆU VỀ XÁC THỰC TIẾNG NÓI

1.1 Tổng quan

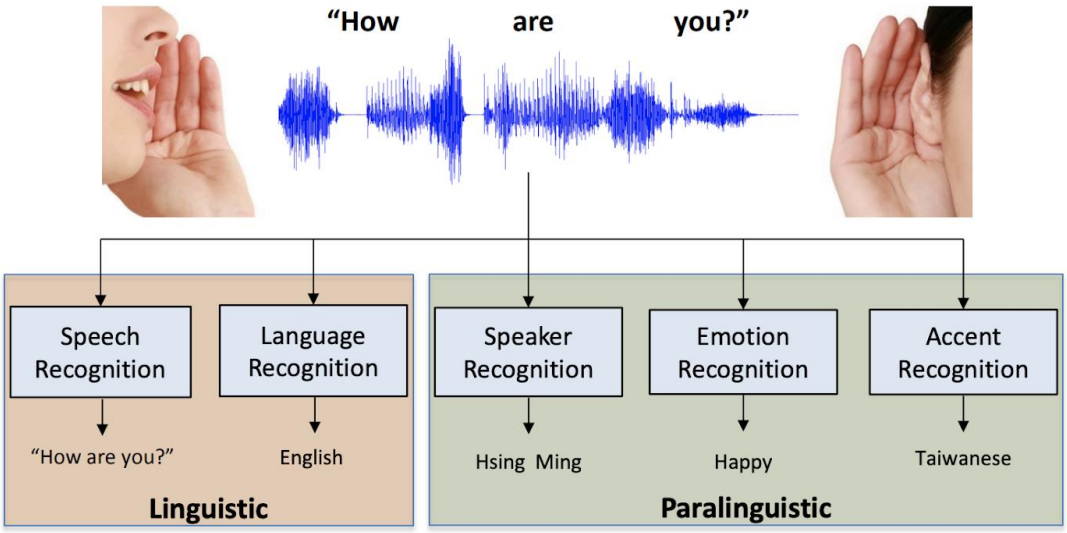
Ở phần giới thiệu chúng ta sẽ zoom rộng ra một chút để nhìn và hiểu rõ hơn vào các bài toán trong trích rút thông tin từ speech.

Trích rút thông tin từ speech được chia ra làm 2 nhóm. Nhóm liên quan đến ngôn ngữ và nhóm không liên quan đến ngôn ngữ.

Nhóm liên quan đến ngôn ngữ thì ta tập trung vào cái mà người nghe đang nói. Những bài toán có thể kể tới như là như là ngôn ngữ như là nhận dạng ngôn ngữ, nhận dạng những từ mà một người đang nói, hay là việc phát hiện người nói nói sai/ thiếu từ hay không.

Còn nhóm không liên quan đến ngôn ngữ thì ta chỉ tập trung vào các đặc điểm như là cảm xúc, chất giọng của người nói và bỏ qua ngôn ngữ nhiều nhất có thể, những bài toán có thể kể tới là phát hiện cảm xúc, phát hiện giọng nói là nam hay nữ...

Bài toán Speaker Verification thì thuộc nhóm không phụ thuộc ngôn ngữ

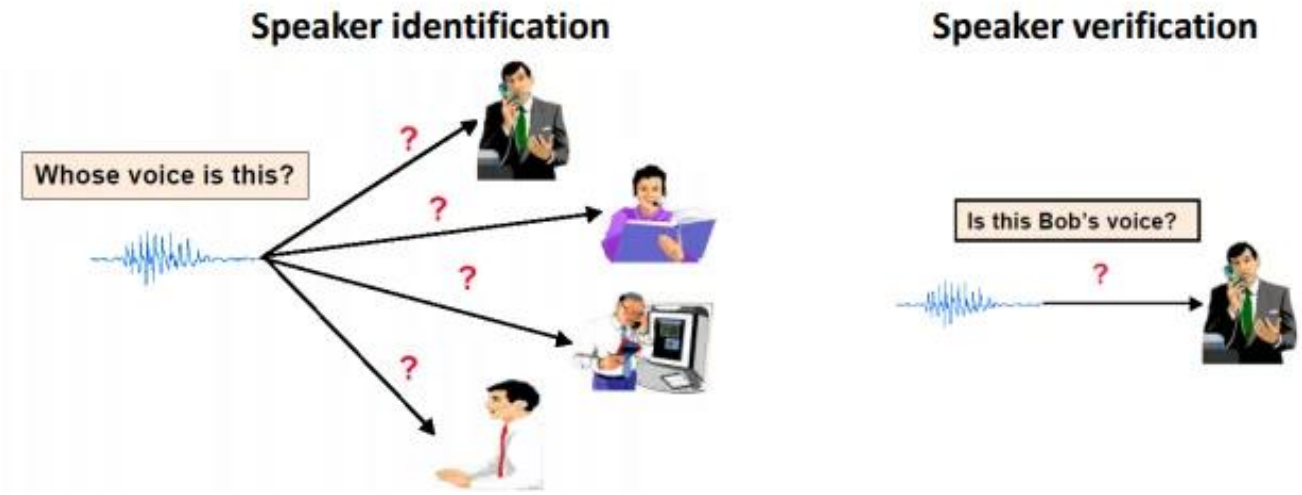


- Speech conveys several types of information
  - Linguistic: message and language information
  - Paralinguistic : emotional and physiological characteristics

Nhóm bài toán speaker verification thì có một vài bài toán liên quan như là

- speaker identification: đầu vào hệ thống là một giọng nói và ta đi tìm người trong cơ sở dữ liệu mà giống với giọng nói ấy nhất.
- ***speaker verification: đầu vào là giọng nói của 1 người và danh tính của một người. Ta sẽ confirm là giọng nói và danh tính này có phải là một người hay không?***
- ***Bài toán thứ 3 là phân đoạn âm thanh: mục tiêu của mình sẽ là phân đoạn âm thanh gốc ra thành các đoạn mà mỗi đoạn chỉ có 1 người nói.***

3 bài toán này rất là gần nhau và 1 cái advance trong bài toán này thì cũng kéo theo advance trong bài toán kia. Cái core thì là representation của audio



Về phương pháp, Speaker Verification cũng được chia thành tiếp cận dựa trên văn bản, với mật khẩu cố định (text-dependent with fixed passwords) và tiếp cận không phụ thuộc vào văn bản (text-independent with no specific passwords) nghĩa là ta sẽ chọn xác nhận dựa trên một số keyword nhất định hoặc không cần keyword.

1.2 Dữ liệu

Do bài toán là bài toán đặc thù về speech do đó khi ta càng nâng cao chất lượng của dữ liệu lên đồng nghĩa với việc mô hình sẽ càng trở nên tốt hơn. Việc build ra một kho dữ liệu sẽ khá khó khăn, ta cần phải thu thập dữ liệu giọng nói của người với dữ liệu sạch. Sạch ở đây có thể hiểu là dính ít tiếng ồn hoặc dữ liệu gồm các utterances một người nói (do khi utterances nhiều người nói thì mô hình sẽ trở nên bị sai do khó khăn trong việc ánh xạ utterances nhiều người nói cho một người). Project này nhắm tới việc build một bộ dữ liệu gồm các utterances chỉ 1 người nói.

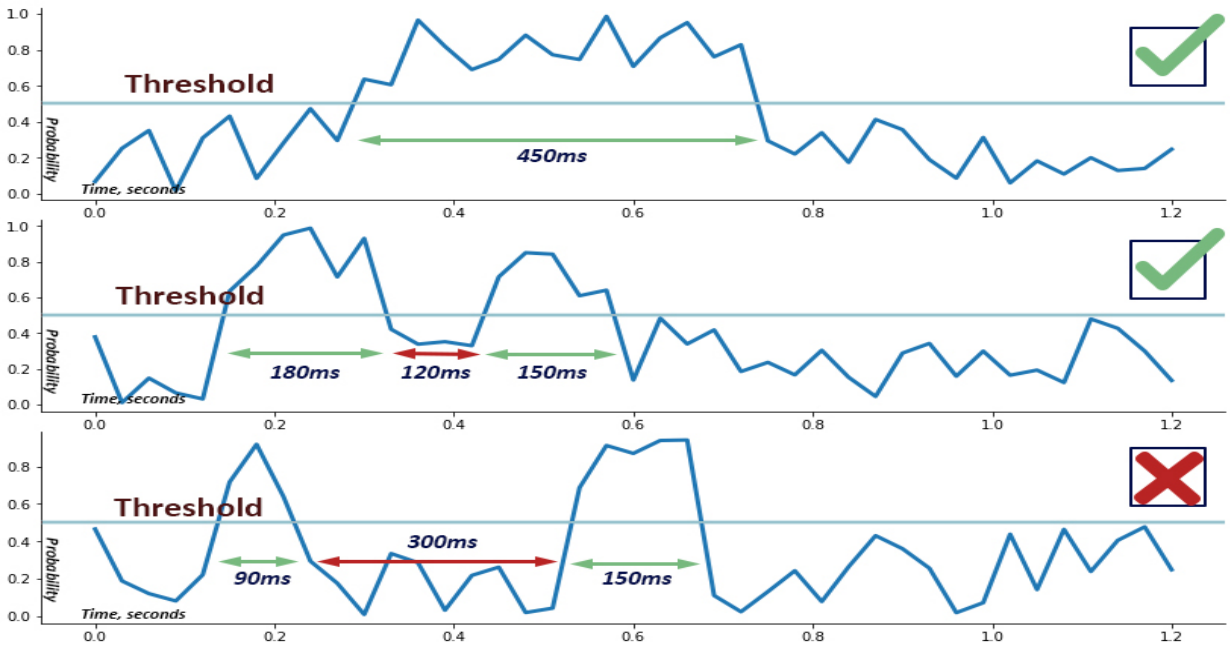
CHƯƠNG 2: GIỚI THIỆU VỀ HAI PRE-TRAINED ĐƯỢC SỬ DỤNG TRONG PROJECT

Ở trong project, chúng ta sử dụng 2 pre-trained model. Một là pre-trained silero VAD được publish tại: [snakers4/silero-vad: Silero VAD: pre-trained enterprise-grade Voice Activity Detector, Language Classifier and Spoken Number Detector (github.com)]

Thứ hai là pre-trained X-vector (ECAPA-TDNN) được publish tại: [speechbrain/speechbrain: A PyTorch-based Speech Toolkit (github.com)]

2.1 Voice activity detection (Vad)

Voice activity detection - phát hiện hoạt động giọng nói là phát hiện sự hiện diện hoặc vắng mặt của lời nói của con người, được sử dụng trong xử lý giọng nói. Ta có thể hiểu đơn giản là trong một đoạn ghi âm hội thoại, Vad sẽ tách được những phần có người nói với những phần im lặng ra. Hình dưới đây mô tả cơ bản về Vad:

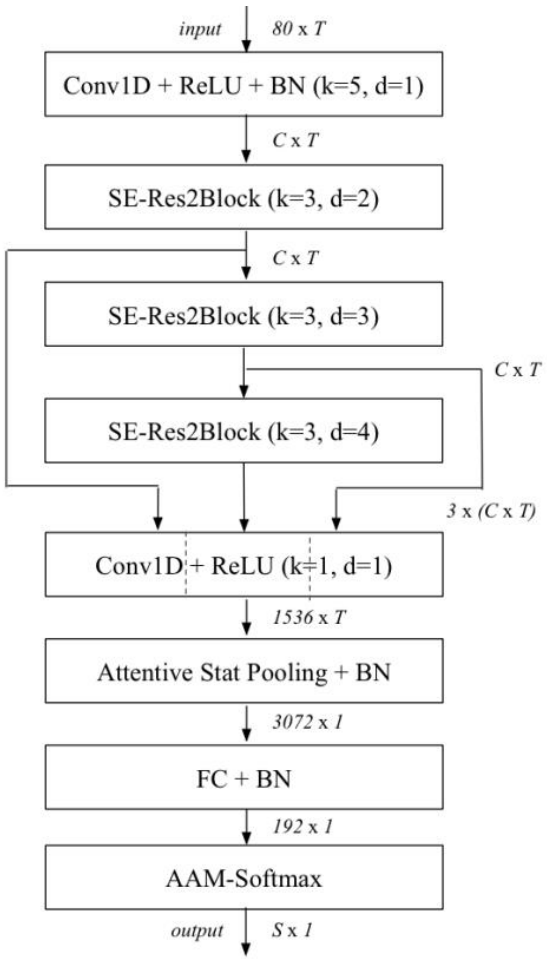


2.2 X-vector

Trong task SV, để phân biệt được lời nói có phải là của nhãn này hay không thì ta cần phải trích xuất được những đặc trưng cơ bản trong tiếng nói của một người. Những đặc trưng này có thể là độ cao, âm sắc, cách luyến láy khi nói.

Nếu Vad sử dụng để phát hiện sự hiện diện giọng nói trong một utterance thì X-vector được sử dụng để trích xuất những đặc trưng giọng nói của utterance đó.

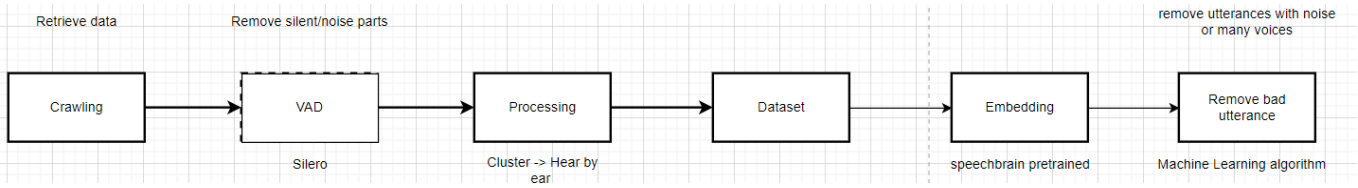
Các đặc trưng được trích xuất này sẽ được biểu diễn dưới dạng vector.



CHƯƠNG 3: PIPELINE, CÁC THUẬT TOÁN ĐƯỢC SỬ DỤNG, CÁC BƯỚC XỬ LÝ DỮ LIỆU VÀ HUẤN LUYỆN DỮ LIỆU VÀ KẾT QUẢ

3.1 Pipeline

Sau đây là pipeline ta sử dụng để build:



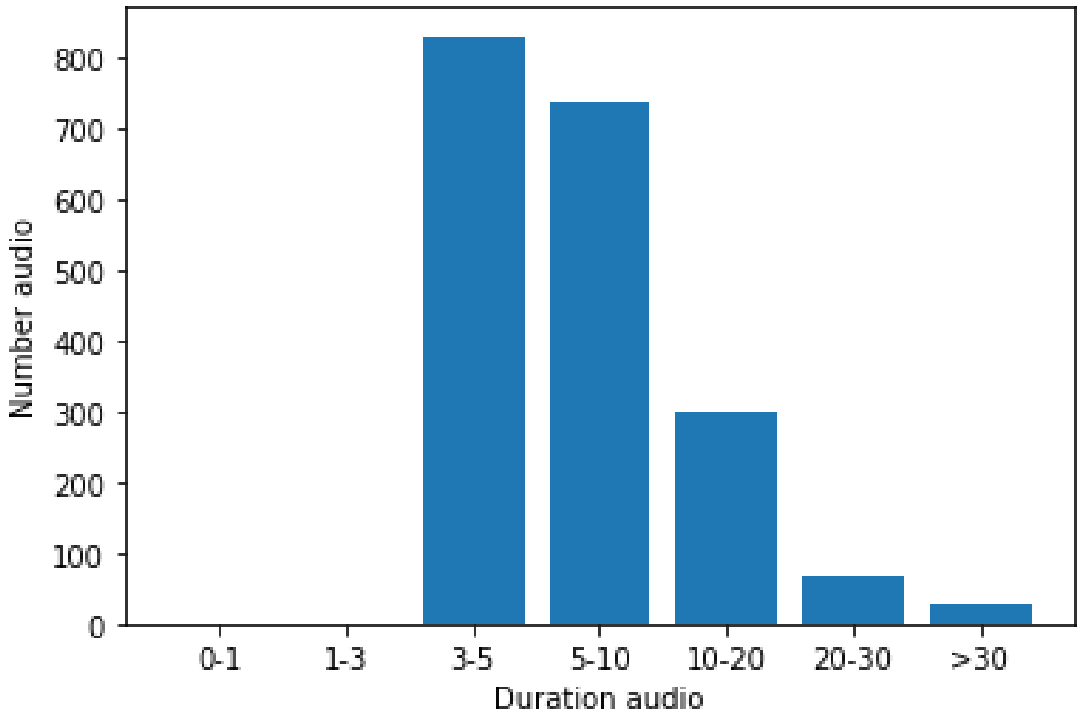
Đầu tiên là ta sẽ crawl dữ liệu (video) từ youtube về. Sau đó các dữ liệu video này sẽ được chuyển về dạng wav. Sau đó những dữ liệu này qua VAD để tách bỏ những dữ phần im lặng/ nhiều tiếng ồn ra, từ đó ta sẽ tách được thành những file nhỏ 3-10s. Sau đó những file nhỏ này được đánh nhãn (để phục vụ việc train dữ liệu đánh nhãn). Những file này sau đó sẽ được qua embedding và được sử dụng thuật toán ML cơ bản để xóa bỏ noise (file mà ồn hoặc là nhiều người nói). Sau khi hoàn thành pipeline, ta có thể áp dụng model này với nhiều video youtube khác nhau từ đó build được dữ liệu.

3.2 Tiền xử lý

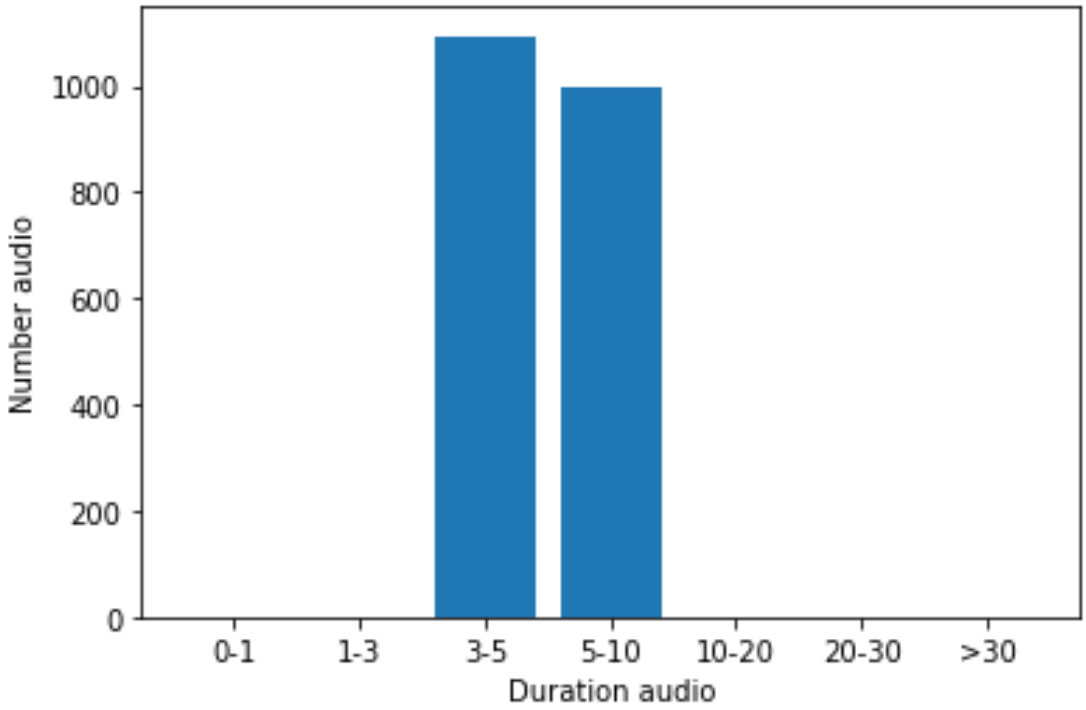
Đề crawl dữ liệu youtube về, đầu tiên, chúng em sử dụng pytube để down dữ liệu từ youtube về, dữ liệu được down lấy từ chương trình “12 con giáp”. Dữ liệu được lấy từ gameshow để đa dạng được người nói. Sau đó xử dụng moviepy để chuyển từ những file mp4 về mp3. Tiếp tục sử dụng pydub để convert những video về

dạng wav để dễ xử lý. Những wav này được lưu dưới sample rate 16000. Tổng thời lượng 3 video crawl về là 6 tiếng.

Sau khi crawl dữ liệu từ youtube về. Chúng ta xử dụng pre-trained silero để VAD những dữ liệu đã có. Những dữ liệu này qua pre-trained silero với threshold là 0.5 (Hay có thể nói xác suất tiếng nói 0.5). Do ta lấy xác suất 0.5 khá là nhỏ nên dữ liệu sẽ được phân bố như sau:



Do dữ liệu >10s khá là nhiều do đó chúng em tiếp tục cho những dữ liệu >10s này qua tiếp VAD silero với threshold 0.9 để cắt được thành những dữ liệu nhỏ hơn. Sau khi cho qua VAD threshold 0.9 thì ta có được dữ liệu phân bổ như sau:



Tổng thời lượng ta thu được là hơn 3 tiếng dữ liệu.

Sau khi VAD ta đã thu được 2089 file dữ liệu 3-10s.

3.3 Xử lý dữ liệu, huấn luyện dữ liệu

3.3.1 Cách tiếp cận thứ nhất

Từ dữ liệu VAD ở trên, ta sử dụng X-vector embedding từ ECAPA-TDNN lên từng file dữ liệu được các mảng dữ liệu tương ứng với mỗi audio, mỗi dữ liệu tương ứng một mảng shape (192,). Ta chia train test thành 2/3 tập train và 1/3 tập test sau đó sử dụng KNN - uniform với K chạy từ 1 đến 20 và sử dụng LR lên dữ liệu train với loss MSE ta được loss như sau:

```
k = 1 Mean Squared Error =  0.2710144927536232
k = 2 Mean Squared Error =  0.21159420289855072
k = 3 Mean Squared Error =  0.1959742351046699
k = 4 Mean Squared Error =  0.18197463768115943
k = 5 Mean Squared Error =  0.18156521739130435
k = 6 Mean Squared Error =  0.1816425120772947
k = 7 Mean Squared Error =  0.18095238095238095
k = 8 Mean Squared Error =  0.1782608695652174
k = 9 Mean Squared Error =  0.17659688674181429
k = 10 Mean Squared Error =  0.174768115942029
k = 11 Mean Squared Error =  0.17699125643789673
k = 12 Mean Squared Error =  0.1792572463768116
k = 13 Mean Squared Error =  0.18074779178458109
k = 14 Mean Squared Error =  0.1821724341910677
k = 15 Mean Squared Error =  0.18359420289855075
k = 16 Mean Squared Error =  0.18674705615942028
k = 17 Mean Squared Error =  0.18822024973672333
k = 18 Mean Squared Error =  0.19092860976918952
k = 19 Mean Squared Error =  0.19240836645389217
k = 20 Mean Squared Error =  0.19293840579710145
```

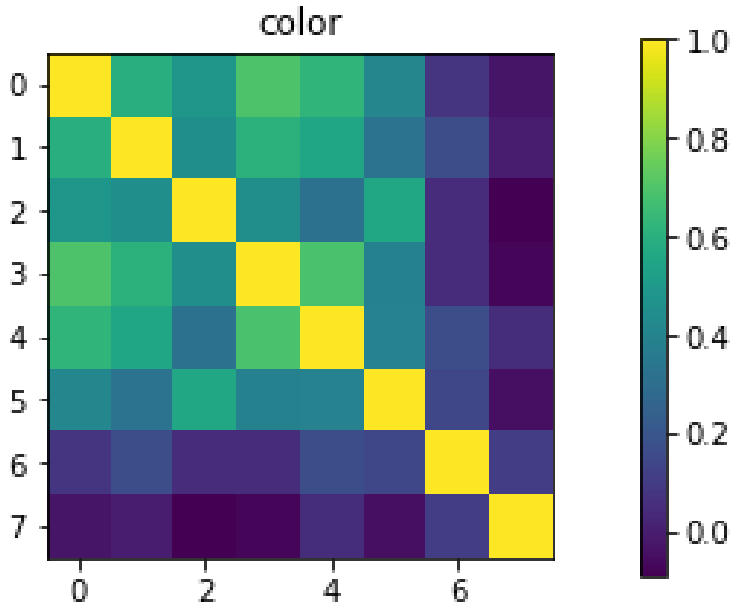
Với độ chính xác 75,78% cho KNN khi K=10 và 70.43% cho LR khi test trên tập test.

3.3.2 Cách tiếp cận thứ hai

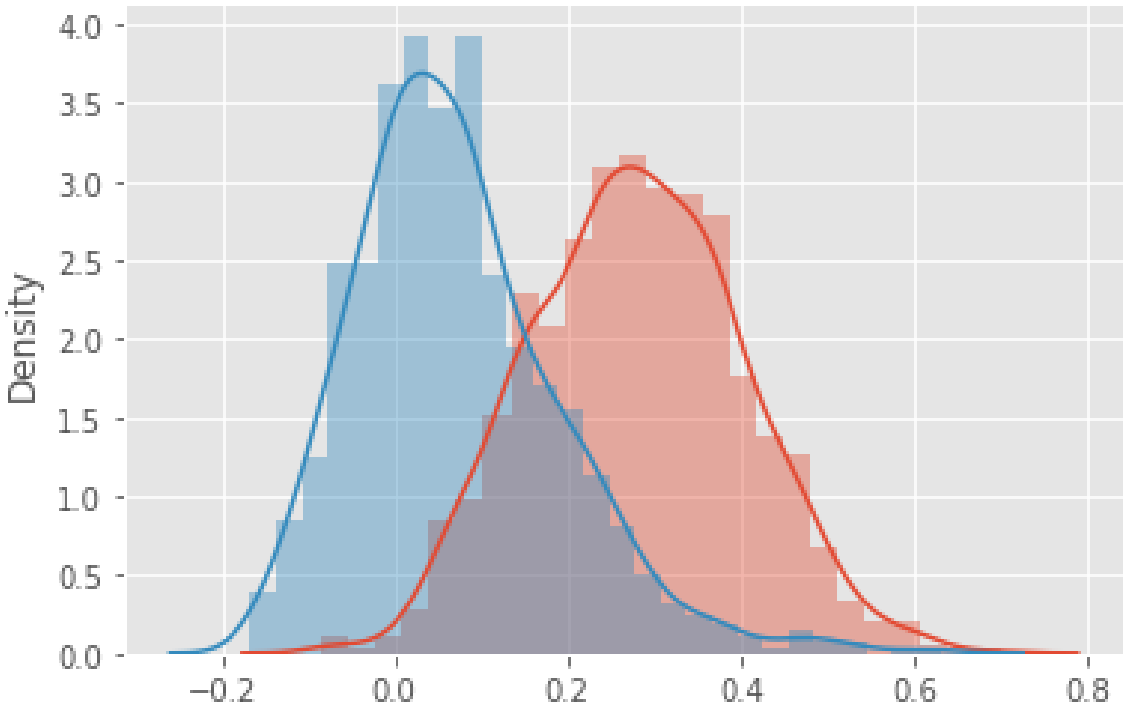
Với cách tiếp cận thứ nhất khá đơn giản và sử dụng trực tiếp thuật toán KNN LR lên data base ta hi vọng có thể làm tốt hơn. Do đó ta đến với một cách tiếp cận khác cho bài toán này.

Đầu tiên với mỗi utterance, ta sẽ chia thành những đoạn có length là 1.2s và overlap là 0.2s. Lý do ta lấy overlap là vì khi ta nói, những âm thanh bên cạnh nhau nó có thể liên quan tới nhau. Sau đó với mỗi đoạn 1.2 ta sẽ sử dụng embedding X-vector ECAPA-TDNN lên nó. Sau đó ta sử dụng cosine_pair lên các đoạn chia của từng audio một. Ta sẽ lấy min cosine_pair đó. Do với X-vector

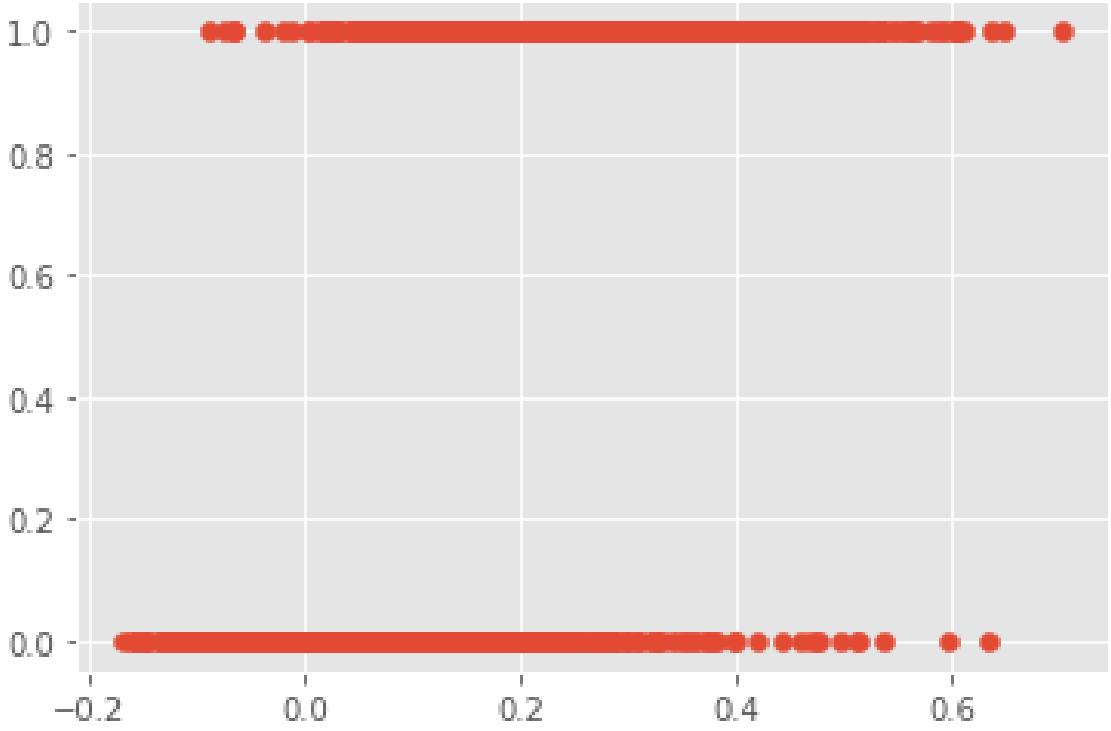
embedding, sự giống nhau giữa các người nói có thể tính bằng cos, khi đó nếu cosine càng cao tức là khả năng là cùng một người càng cao. Do vậy ta lấy min cosine là để tìm phần chênh lệch nhau nhất trong cùng một utterance. Khi đó việc phân chia data sạch thì sẽ sử dụng dựa trên min cosine của từng utterance. Dưới đây là biểu diễn min cosine của một utterance



dụng dựa trên min cosine của từng utterance. Dưới đây là biểu diễn min cosine của một utterance. Sau khi sử dụng min cosine lên các audio. Ta sẽ ra một file cosine và nhãn của nó. Dữ liệu này sẽ phân thành 2 phân phối chuẩn về 2 bên label 0 và 1 với 1 là data sạch thì cosine min sẽ cao còn 0 thì cosine thấp



Biểu diễn lại phân phối



Từ đây ta dễ dàng thấy rằng ta cần chọn một threshold nằm giữa khoảng overlap giữa label 0 và label 1 để phân ranh rới được những data sạch.

Ta áp dụng SVM, KNN và LR vào min cosine và label thu được kết quả như sau:

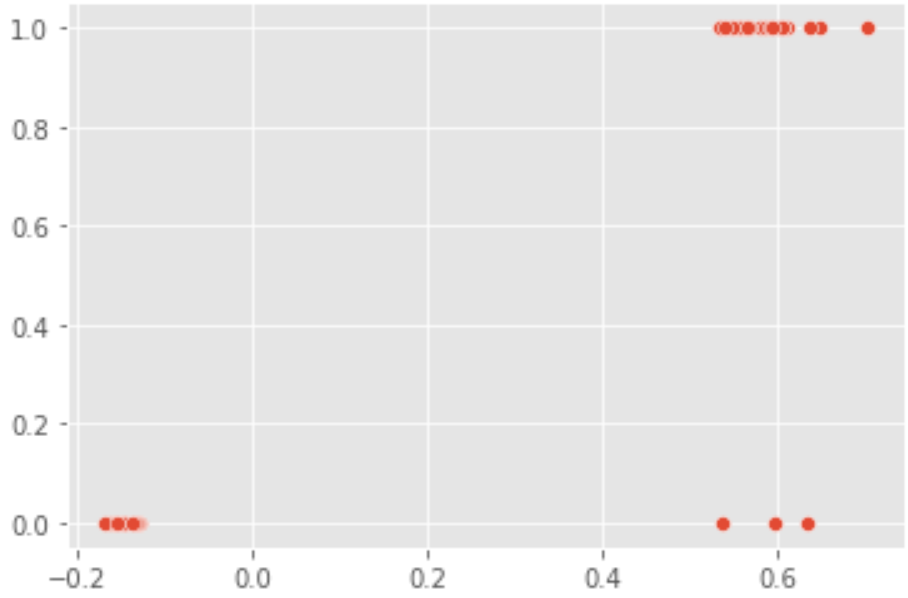
```
k = 1accur = 75.21739130434783%
k = 2accur = 74.63768115942028%
k = 3accur = 80.57971014492753%
k = 4accur = 80.8695652173913%
k = 5accur = 80.72463768115942%
k = 6accur = 81.73913043478261%
k = 7accur = 82.02898550724638%
k = 8accur = 82.6086956521739%
k = 9accur = 82.46376811594203%
k = 10accur = 82.46376811594203%
k = 11accur = 82.6086956521739%
k = 12accur = 82.17391304347827%
k = 13accur = 82.89855072463767%
k = 14accur = 82.89855072463767%
k = 15accur = 83.18840579710145%
k = 16accur = 83.33333333333334%
k = 17accur = 83.33333333333334%
k = 18accur = 83.6231884057971%
k = 19accur = 83.47826086956522%
k = 20accur = 83.18840579710145%
```

Đạt độ chính xác 83.6% với KNN khi K=18 và 83.04% với SVM khi threshold là 0.123 còn LR là 83.2% khi threshold là 0.104.

Sau khi hoàn thành việc thử nghiệm model, ta thử sử dụng LOF để tìm điểm ngoại lai của mô hình này.

3.3.3 Local outlier Factor

Sau khi hoàn tất việc train model, ta thử sử dụng một thuật toán tên là Local outlier Factor (LOF) để thử kiểm tra những điểm ngoại lai trong mô hình.Các điểm ngoại lai được xác định dựa trên khái niệm về mật độ cục bộ (địa phương), trong đó điểm địa phương được cho KNN, với khoảng cách được sử dụng để ước tính mật độ. Bằng cách so sánh mật độ cục bộ của một đối tượng với các mật độ các điểm lân cận, người ta có nhận diện các vùng có mật độ tương tự nhau, và các điểm có mật độ tương đối thấp hơn các điểm lân cận của nó. Các điểm này được gọi là các điểm ngoại lai (outlier)



Sau khi sử dụng điểm ngoại lai ta được 42 điểm ngoại lai với đa số đúng nhãn. Nhưng có 3 điểm đặc biệt khi cosine cao nhưng lại có nhãn 0. Khi check qua thì thấy rằng 3 data đặc biệt đó sẽ chứa toàn nhạc. Do đó ta có thể suy đoán rằng do nhạc này chơi cùng loại nhạc cụ như nhau đều đều cho cả đoạn audio. Do đó mô hình không thể phân biệt được. Đây cũng chính là điểm yếu của mô hình

CHƯƠNG 4: TỔNG KẾT

Trên đây là một số mô hình cơ bản nên xác suất đúng dữ liệu "sạch" có thể chưa được hoàn toàn tối ưu. Mô hình có độ chính xác lớn nhất hiện là KNN với min cosine pair với độ chính xác là 83.6%. Từ các kết quả thu được cho thấy. Với mỗi kiến trúc mô hình khác nhau, cho ta một độ chính xác thu được khác nhau đáng kể (Từ 70% LR lên 75% KNN thường và cuối là 83.6% với KNN cosine pair). Qua quá trình thực hiện project, ta đã hiểu kỹ hơn về việc xử lý, huấn luyện model cũng như các kỹ thuật cải tiến model và phân tích dữ liệu. Trên đây là tất cả những gì bọn em đã tìm hiểu trong quá trình làm đồ án môn học.

TÀI LIỆU THAM KHẢO

Các thư viện python được sử dụng trong bài

Local Outlier Factor (LOF) — Algorithm for outlier identification | by Vaibhav Jayaswal | Towards Data Science

Nhận dạng người nói: một số phương pháp kỹ thuật - VinBigdata - Blog

speechbrain/speechbrain: A PyTorch-based Speech Toolkit (github.com)

snakers4/silero-vad: Silero VAD: pre-trained enterprise-grade Voice Activity Detector, Language Classifier and Spoken Number Detector (github.com)