Universidade Vigo

Utilización de características perceptivas para a clasificación de sinais acústicas

Catuxa Seoane Botana

Traballo de fin de grao
Escola de Enxeñaría de Telecomunicación
Grao en Enxeñaría de Tecnoloxías de Telecomunicación

Titor
Manuel Ángel Sobreira Seoane

Curso 2020/2021

Índice

1. Introdución

Nos últimos anos estamos asistindo a unha reforma radical na forma de entender os procesos industriais: a combinación eficiente de novas técnicas de produción, xunto coa incorporación de tecnoloxías dixitais e intelixencia artificial (IA), deixaron paso ao que se chama **Industria 4.0** [4]. Un dos procesos que se benefician disto é o Control de Produto a Fin de Liña (***End of Line Quality Control – EOL***). No presente proxecto indágase na posibilidade de utilizar a clasificación automática para avaliar a calidade acústica de pezas en fin de liña.

Hoxe en día, os sistemas de detección e clasificación de eventos son suficientemente eficientes como para extraer diferentes propiedades sonoras e detectar eventos acústicos aislados con bastante precisión, mais esta precisión é insuficiente en ambientes ruidosos e superpostos [3]. Un exemplo disto é a utilización dos coeficientes de *MFCC*. O uso destes coeficientes procede do recoñecemento de voz. Os primeiros traballos de clasificación e detección de eventos acústicos incorporaron características utilizadas neste ámbito sen ter en conta as particularidades de cada tipo de son a detectar. Unha das desvantaxes do uso dos *MFCCs* é a baixa robustez ao ruído de fondo. Isto fai descartar esta característica para casos coma a detección de eventos a fin de línea de producción, debido ao alto nivel de ruído de fondo dos entornos industriais. Estudos recentes afirman que a investigación detallada sobre cada unha das características e o seu comportamento con sinais de acústicas ou vibratorias, axudaría a elexir as propiedades correctas para cada aplicación en particular [17]. Neste proxecto plantéxase como alternativa a utilización de características relacionadas ca percepción, seguindo a idea de que se é posible escoitar diferenzas entre diferentes eventos acústicos, estes son susceptíbeis de ser detectados automaticamente.

A psicoacústica é a ciencia que estuda a relación entre as propiedades físicas do son e a sensación auditiva que provoca. Aínda que as características que conforman esta ciencia son de carácter subxectivo, existen métodos de procesado para acadar aproximacións numéricas das mesmas. Un exemplo destas poden ser a sonoridade (*Loudness*), a nitidez (*Sharpness*) ou a rugosidade (*Roughness).*

O traballo comezará cun estudo da aplicación de distintas características perceptivas na detección de eventos e cunha avaliación do entorno de programación para a elaboración dun clasificador. Seguidamente, elaborarase unha análise previa en **Matlab** das características no campo de control da calidade a fin de liña de produción. Finalmente, explicarase a elaboración do clasificador de audio en **Python** e expoñeranse os resultados e conclusións acadadas.

2. Obxectivos

Os obxectivos principais de este proxecto son:

- A realización dun clasificador de son mediante as características de *Loudness*, *Sharpness* e *Roughness*.

- A análise dos resultados para avaliar a efectividade das características mencionadas na clasificación de sinais acústicas, en especial na clasificación da calidade das pezas ao remate da liñade produción *End of Line Quality Control* (EOL).

3. Metodoloxía

Para chegar aos obxectivos nomeados, seguiuse unha metodoloxía baseada nos seguintes puntos:

3.1. Avaliación da separabilidade de eventos mediante a utilización de características psicoacústicas

O estudos máis actuais utilizan comunmente características coma os Coeficientes Cepstrais nas Frecuencias de Mel (MFCCs) ou os patróns Binarios Locais (LBPs), máis non tantos estudos analizan a utilizade das propiedades psicoacústicas na separabilidade de eventos.

O artigo *Features for Audio Classification* [6] realiza unha comparación das diferenzas entre utilizar distintos conxuntos de características, incluíndo un conxunto formado por *Loudness*, *Sharpness* e *Roughness*. A base de datos de audios a clasificar consiste en música clásica, música popular (todos os estilos, excepto clásico), xente falando, alboroto de multitudes (aplausos e berros) e ruído de fondo. A precisión da clasificación xeral deste conxunto é do 84 %, o que leva a pensar que parte de esas características poden chegar a ser válidas na diferenciación sonora, ou chegar a mellorar os métodos xa existentes.

A elección das tres propiedades utilizadas neste traballo baseouse no artigo mencionado e no interés en realizar un estudo da aportación das características nomeadas na separabilidade de eventos.

3.2. Avaliación da posibilidade de utilizar Python no desenvolvemento dun clasificador en tempo real

Existen múltiples posibilidades na elección da linguaxe e entorno de programación para a realización dun clasificador baseado en *Machine Learning* (ML). Finalmente, aínda que parte do traballo se levará a cabo en **Matlab**, empregarase **Python**, xa que consta de numerosas librarías, tanto relacionadas co procesado de son [A.1], coma relacionadas con *Machine Learning* [A.2].

O código deste proxecto realizarase en *Windows*, no IDE (*Integrated Development Environment*) *Visual Studio Code*, utilizando a linguaxe de programación **Python** na versión **3.8.5**.

3.3. Determinación das características perceptivas óptimas para a solución ao problema de clasificación da calidade das pezas ao remate da liña de produción *End of Line Quality Control (EOL)*

Probar unha unidade fabricada ao fin da liña de montaxe é un paso crítico no proceso de produción. É indispensable separar os produtos defectuosos, ou aqueles que non cómpren os límites especificados, das unidades funcionais. A detección das unidades non funcionais é o obxectivo principal da proba.

Nos últimos tempos as probas automáticas sustitúen ás probas máis subxectivas realizadas por operadores humanos. Non obstante, as medicións deben proporcionar uns resultados tan sensibles e completos como os que podería obter un operador humano que utiliza os sentidos auditivos e visuais. Por todo isto, o instrumento de medición debe ter capacidade de aprendizaxe para a correcta valoración dos produtos.

As solucións tradicionais de fin de liña soen encontrar dificultades para adaptarse dende entornos controlados a liñas de produción industrial. Isto moitas veces é debido á presenza de vibracións e altos niveis de ruído xerados polas diferentes máquinas.

Neste traballo realizouse unha avaliación inicial sobre un banco de sons formado por oito audios en formato *WAV*. Os audios que forman este conxunto son gravacións de aperturas e peches dalgunhas pezas de maquinaria industrial. Para cada peche e apertura cóntase co audio pertencente a unha máquina defectuosa e co dunha funcional. Grazas a este traballo previo, pódese intuír como funcionarán as características elixidas na clasificación. Todo o proceso está explicado no apartado 5.

Posteriormente a esta primeira análise, realizarase o desenvolvemento do clasificador, onde tamén se utiliza un banco de sons formado por gravacións dunha ferramenta. Cos resultados acadados tratarase de avaliar se a peza ten unha boa calidade acústica: se a súa emisión é pequena e se a calidade sonora resulta "agradable" en termos de escoita.

4. Fundamentos teóricos

Neste apartado defínense as características perceptivas que se propón avaliar neste traballo.

4.1. *Loudness*

A sonoridade, ou Loudness, é a intensidade que percibimos dun son, é dicir, é unha característica que nos permite ordear os sons de máis 'fortes' a máis 'débiles'. A súa medición é complexa, xa que nela inciden outros factores psicoacústicos.

Mídese mediante unha unidade chamada **Sonios** (*sones*). Esta unidade é a asignación numérica da forza dun son de acordo ca resposta de oíntes cunhas características auditivas normais. Un sonio corresponde ao *Loudness* producido nun tono de 1000 Hz, cun nivel de presión sonora de 40 dB.

Outro concepto importante son as ***Curvas de igual Sonoridade***, polas que se calcula a relación entre a frecuencia e a intensidade (en dB) de dous sons para que sexan percibidos como igualmente altos polo oído, de xeito que todos os puntos da mesma curva teñan o mesmo valor de *Loudness*.

Un Fonio é unha unidade de medida logarítmica e adimensional que serve para indicar o *Loudness* co que se percibe un son. Os **Fonios** (*phons*) son a asignación numérica que acompaña á *Curva de igual Sonoridade*. Se un son determinado percíbese tan alto coma un son de 60dB a 1000Hz, dise que ten un volume de 60 fonios.

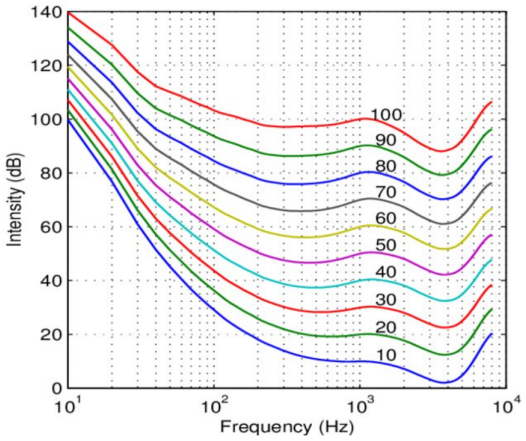


Figura 1: Curvas de igual sonoridade

Neste estudio o valor do *Loudness* ríxese pola norma **ITU-R BS.1770-4** [1]. Polo tanto a súa ecuación durante o intervalo de duración T é:

$$L_K = -0{,}691 + 10 \log_{10} \sum_i G_i z_i \tag{1}$$

sendo G_i os coeficientes de ponderación para as canles individuais e z_i a potencia, calculada coma:

$$z_i = \frac{1}{T} \int_0^T y_i^2 \mathrm{d}t \tag{2}$$

4.2. *Sharpness*

A nitidez, ou Sharpness, é a medida numérica da sensación baseada na cantidade de compoñentes de alta frecuencia nun son. Mídese mediante unha unidade chamada "acum". O valor 1 "acum" asígnaselle a un ruído de banda estreita a 1 kHz, cun ancho de banda menor que 150 Hz e un nivel de 60 dB.

A norma estandarizada para o cálculo do *Sharpness* é **DIN 45692** [2]. Baseándose nesta norma o seu cálculo realizaríase da seguinte maneira:

$$Sharpness = k \left(\frac{\int_{z=0}^{24} N'(z) g(z) z \mathrm{d}z}{\int_{z=0}^{24} N'(z) \mathrm{d}z} \right) \tag{3}$$

Onde N' é o *Loudness* específico en *Sonios/Barks*, k defínese de tal xeito que un sinal acústico de referencia 1 kHz resulta nunha medida de *Sharpness* de 1 acum e $g(z)$ cumpre o seguinte:

$$g(z) = 1 \qquad \text{para } z \leq 15{,}8 Bark$$

$$g(z) = 0{,}15 e^{0{,}42(z-15{,}8)} + 0{,}85 \qquad \text{para } z > 15{,}8 Bark$$

4.3. *Roughness*

A rugosidade, ou Roughness, é un parámetro que mide o grao de molestia provocado por modulacións rápidas nas sinais acústicas. Mídese mediante unha unidade chamada "asper". Un *asper* defínese como o *Roughness* producido por un ton de 1000Hz de 60dB que é 100 % de amplitude modulada a 70Hz.

Seguindo o modelo do cáculo do *Roughness* do libro *Signal Processing in Acoustics VOL 1* [15], a fórmula do *Roughness* acadaríase da seguinte maneira:

O *Roughness* dun canal específico i, sería:

$$r_i = (g(z_i)m_i^* k_{i-2}k_i)^2 \tag{4}$$

Sendo m_i o grao de modulación, $g(z_i)$ unha función de ponderación con factores de 0,6 a 1,1 segundo a dependencia do *Roughness* da frecuencia portadora dos tonos modulados en amplitude e k os coeficientes de correlación cruzada.

Polo que a fórmula do *Roughness* total é:

$$R = 0{,}25 \sum_{i=1}^{47} r_i \qquad [asper] \tag{5}$$

5. Análise *End of Line Quality Control (EOL)*

O desenvolvemento deste proxecto comezará cun estudo en **Matlab** realizado sobre un banco de sons formado por 8 gravacións de diferentes movementos dunha ferramenta industrial. Os arquivos de audio están en formato WAV e están formados por movementos de apertura e peche da ferramenta en dúas circunstancias diferentes: funcionamento da ferramenta en estado normal e funcionamento da mesma ferramenta en estado defectuoso. A análise realizada baseouse na interpretación das gráficas de *Loudness* e *Sharpness* xeradas sobre cada un dos arquivos de audios para unha previa aproximación do comportamento destas propiedades e se aportan información sobre a calidade sonora das mostras. Posteriormente realizouse o mesmo para o *Roughness* grazas á actualización **Matlab2021 - a**, que implementa novas funcionalidades.

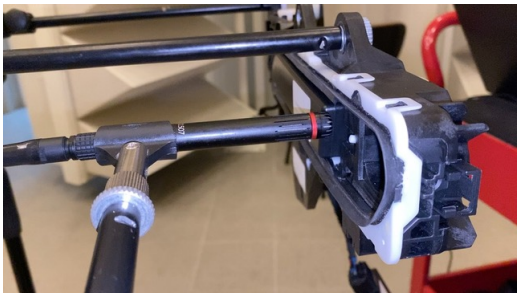


Figura 2: Micrófono utilizado para a gravación das mostras

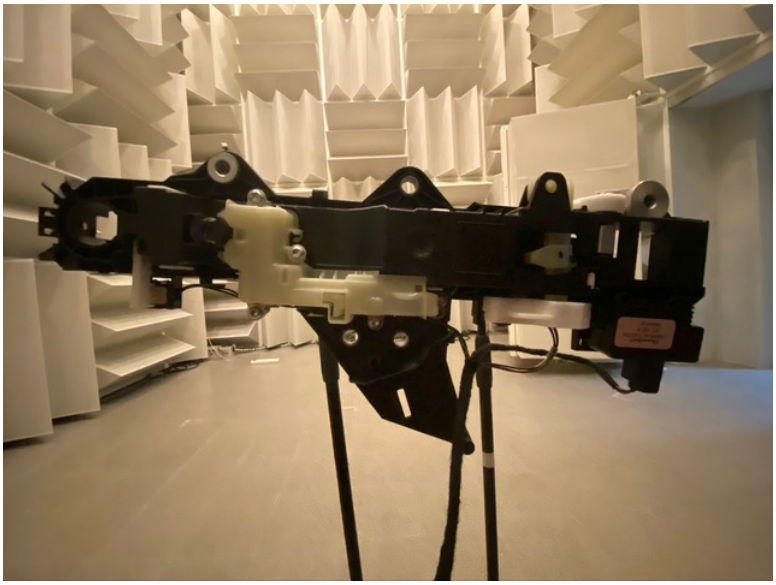


Figura 3: Ferramenta industrial na cámara semianecoica da EET da UVigo

Na seguinte figura móstranse os resultados acadados cun dos movementos de peche da ferramenta no caso da *Loudness*:

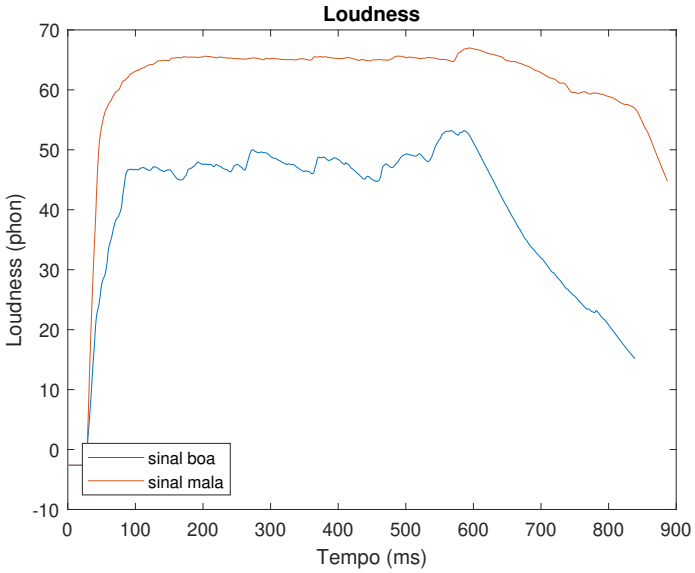


Figura 4: *Loudness* en *phon*/*ms* do peche da ferramenta

Na figura 4 podemos ver unha gran diferenza entre o valor da propiedade das dúas sinais. No caso da ferramenta defectuosa apreciamos claramente un maior *Loudness* en *phon*/*ms*, o que nos indica que a sinal mala percíbese como máis 'forte'

Na seguinte figura móstranse os resultados acadados cun dos movementos de peche da ferramenta no caso da *Sharpness*:

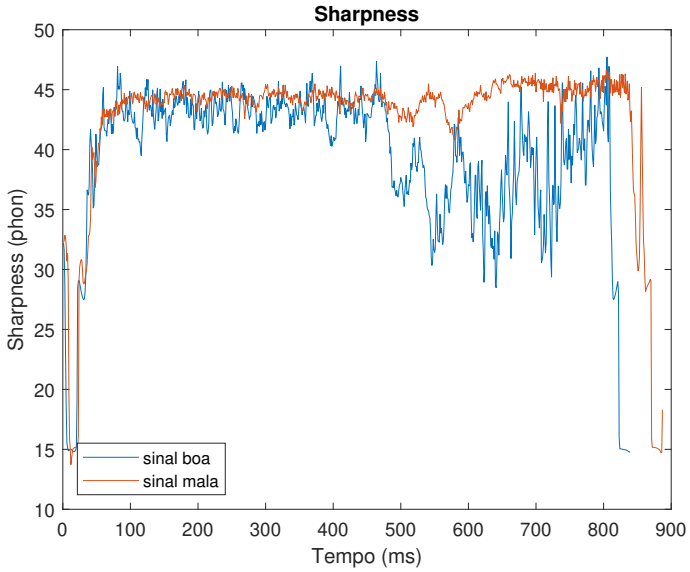


Figura 5: *Sharpness* en *phon/ms* do peche da ferramenta

No caso da figura 5 non se aprecia con claridade unha diferenza entre as dúas sinais no caso da nitidez. Os resultados das dúas sinais son diferentes pero non hai un patrón ao longo do tempo que nos poida servir de utilidade neste caso.

Na seguinte figura móstranse os resultados acadados cun dos movementos de peche da ferramenta no caso do *Roughness*:

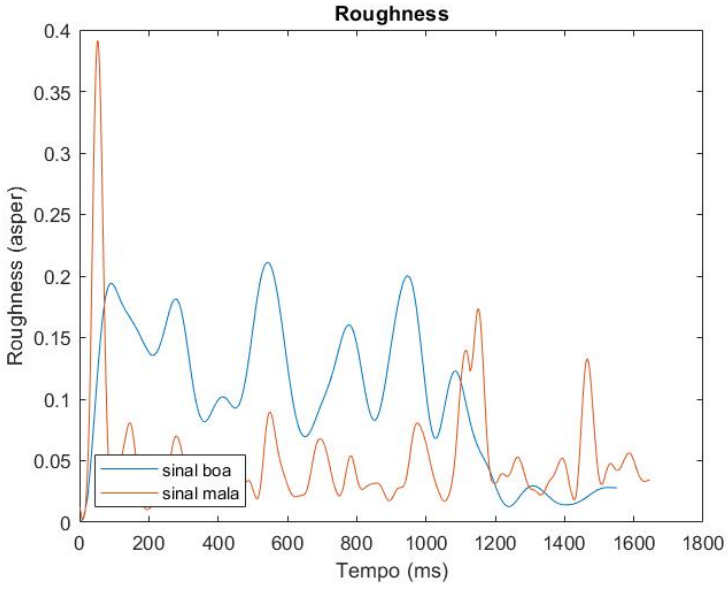


Figura 6: *Sharpness* en *asper* do peche da ferramenta

Na figura 6 pódense ver variacións entre as sinais, pero non cun patrón marcado. Non obstante, pódese observar que nos últimos segundos a variación da característica na sinal mala é moito maior.

6. Desenvolvemento do clasificador de sinais acústicas en Python

O desenvolvemento do código para a separabilidade de eventos acústicos consta dunha parte de adestramento e cunha de predición. O código encóntrase no seguinte enlace: https://github.com/catuxaseoane/Clasificador_ferramentas_Psicoacustica

Contouse cun banco de sons de 81 arquivos WAV; dos cales 54 corresponden a gravacións de maquinaria industrial en bo estado (gardados nunha carpeta chamada *1- audios_GOOD*) e 27 a maquinaria industrial defectuosa (gardados nunha carpeta chamada *1- audios_BAD*). Todos eles teñen unha frecuencia de muestreo de 4800 Hz e unha duración de 400 ms. Este banco de sons foi suministrado ao inicio da elaboraboración do proxecto.

Cadro 1: Conxuntos de arquivos utilizados para a clasificación

Carpeta	Descripción	Número de audios	Segundos
audios_GOOD	Gravacións en formato WAV de maquinaria industrial en estado funcional	54	21.6
audios_BAD	Gravacións en formato WAV de maquinaria industrial en deterioro ou defectuoso	27	10.8

O valor elixido de *test_size* foi do 20 %, isto quere dicir que se utilizaron 64 audios para o adestramento e 17 para o conxunto de probas, que nos permiten saber a calidade dos resultados que obtemos.

Cadro 2: Conxuntos de arquivos

test_size	train_size	TOTAL
17 (20 %)	64 (80 %)	81 (100 %)

Empregouse código do proxecto de código aberto *babycrydetection* [18].

6.1. Etapa de adestramento

A realización do código comezará cunha etapa de adestramento que se dividirá en dúas partes: a extracción das características psicoacústicas e a etapa de clasificación.

6.1.1. Extracción de características

Neste apartado, abordarase o problema de extraer características dunha sinal de audio e o proceso de acadar, a partir dun arquivo *.wav*, unha representación válida para un modelo de *IA*.

A combinación das librarías *Librosa* [A.1.1] e *NumPy* [A.2.2] é unha das maneiras máis doadas de procesado de audio en **Python**, pese á súa tediosa instalación. *Librosa* consta de funcións para a extración de características de audio que devolven estas en *numpyarrays*; que son vectores multidimensionais e matrices con gran capacidade para gran cantidade de datos. Isto facilita o procesado dos datos nas librarías implementadas para *ML*.

Desafortunadamente, *Librosa* non implementa ningunha función que calcule a sonoridade, nitidez ou rugosidade dun audio. Por iso, farase unha modificación de diferentes funcións de librarías de **Python** que si conteñan o calculo destas. Para cada característica levarase a cabo un proceso diferente.

As matrices resultantes da extracción de características, xunto coas etiquetas do tipo de clase (*1- audio_GOOD* ou *2- audio_BAD*) á que pertence cada un dos audios representados, constituirán os *datasets* de adestramento e estarán contidos en diferentes arquivos de formato *.npy (Python NumPy Array File)*.

Extracción do *Loudness*

Para a extracción da sonoridade utilizarase a libraría de código libre *Pyloudnorm* [A.1.2], que se rixe pola norma **ITU-R BS.1770-4** [1]. Dividirase cada arquivo de audio nun total de 24 tramas, utilizando a clase *Meter* cun valor de *block_size* de 60 ms e un solapamento do 75 %. Calcularase o valor de cada trama e introduciranse nunha matriz. As matrices de todos os audios introduciranse nun *numpy.array* que formará a característica *loudness_feat*, que se utilizará posteriormente para o modelo de clasificación.

Extracción do *Sharpness*

Para a extracción da nitidez utilizarase a libraría de código libre *AudioCommons Timbral Models* [A.1.3], que contén funcións para o cálculo da propiedade que se basean na implementación de Fastl de 1991 [14]. Dividirase cada arquivo de audio nun total de 24 tramas, neste caso axustando o valor do parámetro *window_lenght* (tamaño da ventá) en varias das funcións. Calcularase o valor de cada trama e introduciranse nunha matriz. As matrices de todos os audios introduciranse nun *numpy.array* que formará a característica *sharpness_feat*, que se utilizará posteriormente para o modelo de clasificación.

Extracción do *Roughnesss*

Para a extracción da rugosidade utilizarase a libraría de código libre *Mosqito* [A.1.4], que contén funcións para o cálculo da propiedade que se basean na implementación do *Roughness* de Daniel e Weber de 1997 [16]. Dividirase cada arquivo de audio nun total de 24 tramas utilizando un valor de ventá de 16 ms e un solapamento do 0.5 %. Isto realizouse desta maneia xa que no caso de utilizar os valores de ventá maiores, cun maior solapamento, varias funcións da libraría xeraban erros. Posteriormente, calcularase o valor de cada trama e introduciranse nunha matriz. As matrices de todos os audios introduciranse nun *numpy.array* que formará a característica *roughness_feat*, que se utilizará posteriormente para o modelo de clasificación.

6.1.2. Clasificación

A clasificación das características realizarase por medio da libraría *Scikit-learn* [A.2.1]. Utilizarase, a partir desta libraría, o modelo de aprendizaxe supervisado *SVC (Support-vector machine)*. O propósito do aprendizaxe supervisado é que o algoritmo poida "aprender" comparando a súa saída real coas saídas "ensinadas" para atopar erros e modificar o modelo en consecuencia.Por outra parte, para evitar un sobreaxuste dos datos utilízase o método da validación cruzada. Este método implementarase mediante a clase *GridSearchCV*. O esquema de funcionamento desta técnica é móstrase no seguinte esquema:

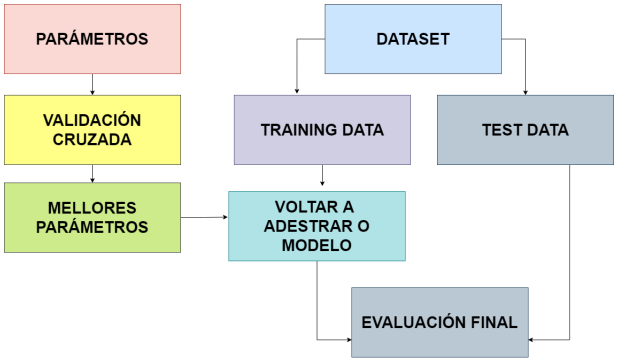


Figura 7: Funcionamento da validación cruzada na avaliación do rendemento dun estimador

Finalmente o modelo resultante gardarase nun arquivo en formato *.pkl (Python Pickle File)*, que será utilizado posteriormente na etapa de predicción.

A clasificación realizouse tanto para un *dataset* formado polas características de *Loudness*, *Sharpness* e *Roughness* coma para as súas múltiples combinacións.

6.2. Etapa de predicción

Nesta etapa do traballo utilízanse as mesmas librarías que as mencionadas na etapa de adestramento [6.1]. O obxectivo desta etapa é clasificar un audio dado mediante o modelo previamente adestrado. No caso de que o resultado da predicción fose que a ferramenta está en bo estado escribiríase un 1 nun arquivo en formato *.txt* chamado prediction.txt, no caso de de que a predicción dose que a ferramenta está nun mal estado, escribiríase un 0.

Sería interesante realizar unha validación dos resultados mediante a predicción de mostras que non foran empregadas no adestramento do modelo, mais a limitación de audios non permitiría uns resultados precisos. Isto poderíase facer con máis audios ou mediante un aumento do *dataset*.

7. Resultados

Os resultados do clasificador acadados utilizando cada unha das características e as súas combinacións móstranse no seguinte cadro:

Cadro 3: Resultados do clasificador de eventos acústicos en Python

Característica	**Acuracy**	**Recall**	**Precisión**	**f1**
Loudness	0.8235	0.8257	0.8071	0.8132
Sharpness	0.7647	0.7424	0.7424	0.7424
Roughness	0.6471	0.5	0.3235	0.3928
Loudness e Sharpness	0.8235	0.8258	0.8071	0.8132
Loudness e Roughness	0.8824	0.8258	0.8071	0.8132
Sharpness e Roughness	0.7059	0.7348	0.7153	0.7018
Loudness, Sharpness e Roughness	0.7647	0.7803	0.7569	0.7571

No cadro 7 podemos ver as métricas de *accuracy*[1], *recall*[2], precisión[3] e *f1*[4] que avalían o rendemento do clasificador en dependencia das características psicoacústicas utilizadas. Estas métricas calcúlanse a partir das matrices de confusión que se atopan no anexo [B.2]

Das tres características, a que mostra uns millores resultados é o *Loudness*, cunha precisión do **80,71 %**. A análise do apartado 5 mostra un exemplo da diferenciación das sinais no caso da sonoridade. Isto, unido aos resultados acadados no clasificador era esperable xa que, cando unha máquina se estropea, soemos percibir o son desta como máis 'forte', o que explicaría o incremento da propiedade e os resultados obtidos na separabilidade das dúas clases de eventos.

A precisión acadada polo *Sharpness* é do **74,24 %**. O *Sharpness* é un parámetro psicoacústico con moita influencia no desagrado de sons, mais esta característica non mostra uns resultados moi precisos nesta clasificación. Esta propiedade ao ser concatenada coas outras dúas non acada uns millores resultados, como se pode ver no cadro 7. Pese a que esta característica ten unha influencia considerable no desagrado (*unpleasantness*) dos sons, non se pode chegar a unha conclusión clara na clasificación deste tipo de ferramenta, mais pode ser posible que isto millore no caso de clasificar maquinaria que emita un son con maior contido espectral.

O *Roughness* ten peor resultado de todas as características e combinacións destas, cunha precisión do **32.35 %**. Nun primeiro momento pode resultar extraño, xa que nas pezas defectuosas o proceso de peche emite un son similar a un "tartamudeo", mostrando así unha sinal en fase final modulada en amplitude; o que levaría a pensar que o *Roughness* podería aportar información. Sen embargo, tendo en conta que esta propiedade baséase en modulacións de envolvente temporal de frecuencia media, só se pode obter unha estimación precisa para tramas de audio relativamente longos ($> 180\ ms$) [6] e os audios utilizados eran dunha lonxitude tan curta que imposibilitaba a correcta clasificación. A pesar de todo isto, pódese observar que o clasificador de *Loudness* millora o seu valor de *accurancy* cando se concatena a característica do *Roughness*; pasa dun valor de *accurancy* do **82.35 %** a un de **88.24 %**, que é o millor resultado de esta métrica. O *accuracy* é o total de acertos sobre o total das prediccións e, según podemos ver nas matrices de confusión [B.2] e [B.2], a estimación das pezas malas millora. De

[1] *accuracy*: Mide a porcentaxe de casos que acertou o modelo.
[2] *recall*: Cantos dos verdadeiros positivos foron acertados.
[3] precisión: Proporción de verdadeiros positivos acertados ao total dos verdadeiros positivos previstos.
[4] *f1*: Combinación da precisión e *recall* nun só valor.

esta maneira, refórzase a explicación de que o *Roughness*, a pesar da falta de precisión, recolle adecuadamente o "tartamudeo" da fase dinal do movemento das pezas defectuosas.

A continuación móstranse os diagramas de dispersión en 2 dimensións resultantes dos 6 primeiros casos mostrados no cadro [7]:

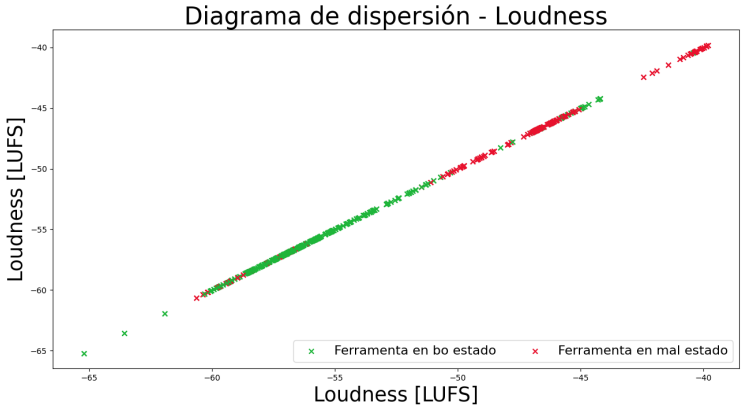


Figura 8: Diagrama de dispersión do *Loudness*

Na figura 8 (Diagrama de dispersión do *Loudness*) podemos ver que entre as ferramentas en bo estado e as ferramentas en mal estado existe unha diferenciación notoria e confírmase a hipótese da análise previa [5] sobre o aumento do *Loudness* nas pezas malas.

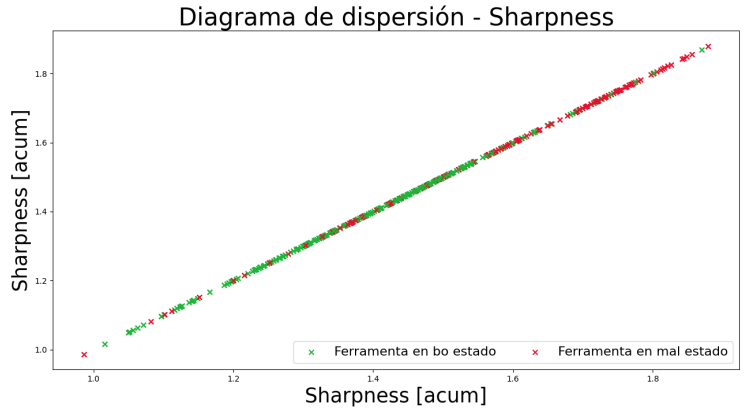


Figura 9: Diagrama de dispersión do *Sharpness*

Na figura 9 (Diagrama de dispersión do *Sharpness*) os resultados non son tan claros como no caso anterior. Aínda así, observando o diagrama e a matriz B.2, podemos observar bastantes acertos no caso dos audios de pezas boas.

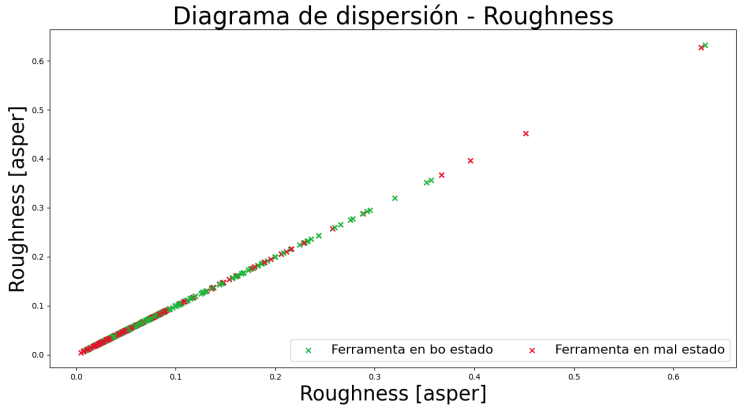


Figura 10: Diagrama de dispersión do *Roughness*

Na figura 10 (Diagrama de dispersión do *Roughness*) os resultados non aportan suficiente información, seguramente debido á falta de segundos de audio no adestramento.

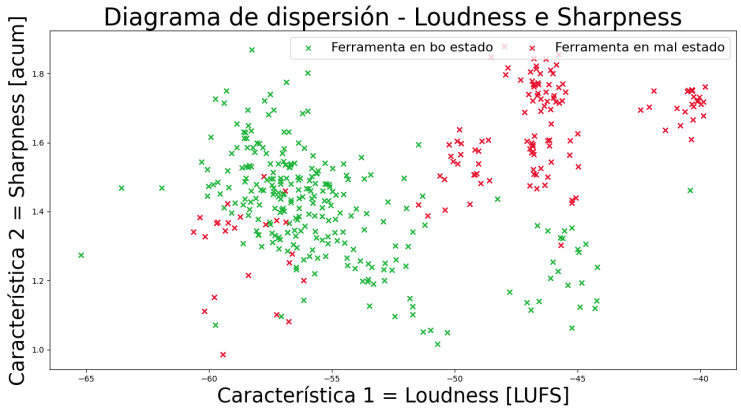


Figura 11: Diagrama de dispersión - *Loudness* vs *Sharpness*

Na figura 11 (Diagrama de dispersión - *Loudness* vs *Sharpness*) pódense observar nubes de puntos divididas en clases separadas, mais a distancia entre puntos da mesma clase é bastante amplia.

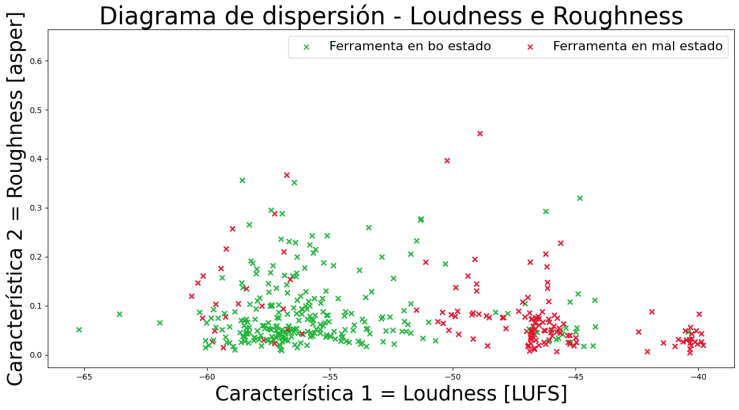


Figura 12: Diagrama de dispersión - *Loudness* vs *Roughness*

Na figura 12 (Diagrama de dispersión - *Loudness* vs *Roughness*) obsérvase o millor resultado de todos os modelos. Os datos das clases *audios_ GOOD* e *audios_ BAD* están diferenciados e os seus respectivos puntos están a distancias cercanas. Isto denota a existencia da relación entre as dúas clases de medicións e reafirma os resultados acadados mediante a matriz de confusión [B.2] .

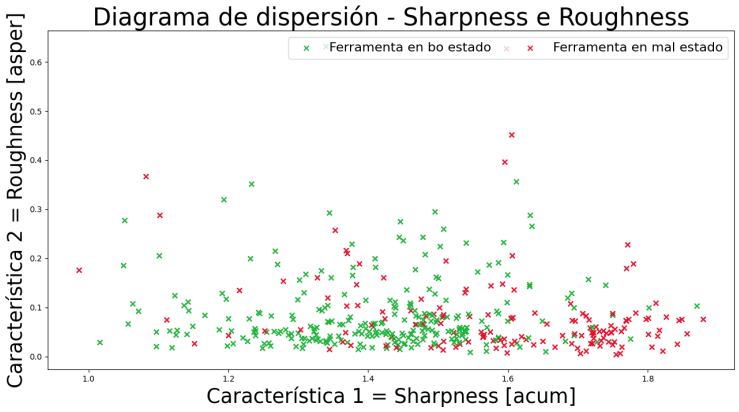


Figura 13: Diagrama de dispersión - *Sharpness* vs *Roughness*

Na figura 13 (Diagrama de dispersión - *Sharpness* vs *Roughness*) obsérvase un escenario moito máis disperso que no caso anterior, polo que, sen un aumento do *dataset* non obteríamos uns resultados claros por medio da concatenación de *Sharpness* e *Roughness*

Finalmente móstrase no cadro 4 de tempos coa que se poderá concluír no apartado seguinte se o clasificador é factible a tempo real:

Cadro 4: Tempos de execución medios (en segundos) do clasificador de sinais acústicas (arquivos de audio de 0.4 segundos)

Tempo de extracción de características/audio	Tempo de predicción/audio	Tempo total de entrenamento
0.0387	0.7915	69.57

8. Conclusións

Neste traballo analizouse a efectividade das características psicoacústicas *Loudness*, *Sharpness* e *Roughness* na separabilidade de eventos acústicos, pero máis concretamente na clasificación da calidade das pezas ao remate da liña de produción.

O banco de sons para o que se contou foi reducido, xa que o número de pezas utilizadas nas gravacións foi moi limitado. Isto debeuse a que as pezas estaban en proceso de diseño. Por esta razón as conclusións sobre os resultados acadados están relativizadas a esta limitación.

Para o caso do *Loudness*, os resultados acadados foron positivos, chegando a unha precisión de clasificación do **80,71 %**. Demostrouse que as pezas en mal estado emiten un son que se percibe como 'forte', o que incrementa a propiedade *Loudness*, e que consigue diferenciarse así das pezas en bo estado. Polo que, concluíndo, esta característica podería ser unha boa opción na clasificación de eventos sonoros no ámbito industrial.

Para o caso do *Sharpness*, o resultado da precisión da clasificación é do **74,24 %**. Por outra parte, a concatenación cas outras dúas características tampouco aporta moita información; xa que non proporciona unha milloría nas métricas xunto ao *Loudness* e, os valores das métricas xunto ao *Roughness* empeoran sutilmente.

Á hora de cuantificar o grao de desagrado dos sons con similares valores dos niveis equivalentes, percibirase unha maior sensación de molestia no que presente un maior contido espectral en altas frecuencias e canto maior é a parte de alta frecuencia dentro dun son, maior é a impresión de *Sharpness* que emite este. Isto quere dicir que esta propiedade aumentará co grao de molestia. Por outra parte, na análise previa [5], os valores desta característica entre as sinais non mostran grandes diferenzas, polo que pódese concluír que o *Sharpness* poderá ofrecer máis información na clasificación de maquinaria que emita sonidos en maior frecuencia. Sería interesante a realización deste estudo con audios pertencentes a outro tipo de ferramentas.

Para o caso do *Roughness*, os resultados do clasificador non acadaron unha precisión moi alta; do **32.35 %**. Isto presuponse que é debido á curta duración dos arquivos de audio, xa que, tendo en conta que esta propiedade baséase en modulacións de envolvente temporal de frecuencia media, só se pode obter unha estimación precisa para tramas de audio relativamente longas ($> 180\ ms$) e as tramas utilizadas para a extracción desta característica foron de 16 ms. Aínda así, coa concatenación desta característica e o *Loudness* millorouse, case nun 6 %, a métrica do *accuracy* resultante de clasificar soamente a sonoridade; a estimación das pezas malas millora. Polo que se demostra que o *Roughness*, a pesar da falta de precisión, recolle adecuadamente o "tartamudeo" da fase final do movemento das pezas defectuosas. Sería interesante o estudo desta característica con gravacións máis extensas e cun maior número de arquivos.

Para avaliar se este clasificador é viable a tempo real a predicción tería que durar menos de un segundo, que é o tempo que se tarda en substituír unha peza para que entre a seguinte a avaliar. Neste caso, tal e como podemos ver no cadro 4, o tempo de predicción para un audio de 0.4 segundos é de 0.7915, polo que isto é posible. Mais, no caso de que se aumente a duración dos audios no adestramento para unha milloría das métricas, teríase que analizar de novo se isto sería posible.

9. Liñas futuras

- Creación dunha libraría específica para EOL en Python que contemple:
  - Traballar con diferentes lonxitudes de audio e tamaños de trama oara adaptar a clasificación ao tipo de problema.
  - Incluir funcións de streaming para traballar con adquisición de audio en tempo real (a versión actual traballa sobre ficheiros)
- Realización de probas sobre as características con conxuntos de sons máis numerosos, máis diversos e cunha maior lonxitude.

Apéndice

A. Estado da arte

A.1. Librarías para a extracción de características en Python

A.1.1. Librosa [7]

Librosa é un paquete de Python que se utiliza para a análise de música e audio. Consta de diferentes funcións que permiten obter múltiple información dunha sinal acústica.

A.1.2. Pyloudnorm [8]

Pyloudnorm é unha libraría de Python coa que se pode obter o *Loudness* medio dunha señal acústica introducida. Permite controlar ogating block size e os filtros de ponderación de frecuencia para un control adicional.

Esta libraría ríxese pola norma **ITU-R BS.1770-4** [1], que é unha recomendación sobre a implementación dos algoritmos para medir o *Loudness* dos programas radiofónicos e o nivel de cresta do audio real.

A.1.3. AudioCommons Timbral Models [9]

Os modelos *Timbral Models* foron desenvoltos polo *Institute of Sound Recording* (IoSR) da Universidade de Surrey e completáronse como parte do proxecto *AudioCommons*.

A distribución actual contén scripts de **Python** para predicir oito características timbrais: dureza, profundidade, brillo, rugosidade, calor, nitidez, *booming* e reverberación.

A.1.4. Mosqito [10]

MOSQITO é un marco de desenvolvemento unificado e modular de ferramentas de calidade de son (incluídas as claves métricas SQ) que favorecen a ciencia reproducible e a escritura compartida eficiente entre a comunidade de enxeñeiros, profesores e investigadores.

A.2. Librarías para *Machine Learning* (ML) en Python

A.2.1. Scikit-learn [11]

Scikit-Learn é unha biblioteca gratuíta para Python. Ten algoritmos para a clasificación, regresión, agrupación e redución de dimensionalidade. Ademais, presenta a compatibilidade con outras bibliotecas Python como NumPy, SciPy e matplotlib. As principais vantaxes de scikit-learn son a facilidade de uso e a gran cantidade de técnicas de aprendizaxe automática que implementa.

A.2.2. NumPy [12]

NumPy é unha biblioteca de Python especializada en cálculo numérico e análise de datos. Incorpora unha clase de obxectos chamados **numpy.arrays**. Estes permiten representar coleccións de datos do mesmo tipo en varias dimensións e funcións moi eficientes para o seu tratamento.

B. Gráficas

B.1. Gráficas da análise previo EOL

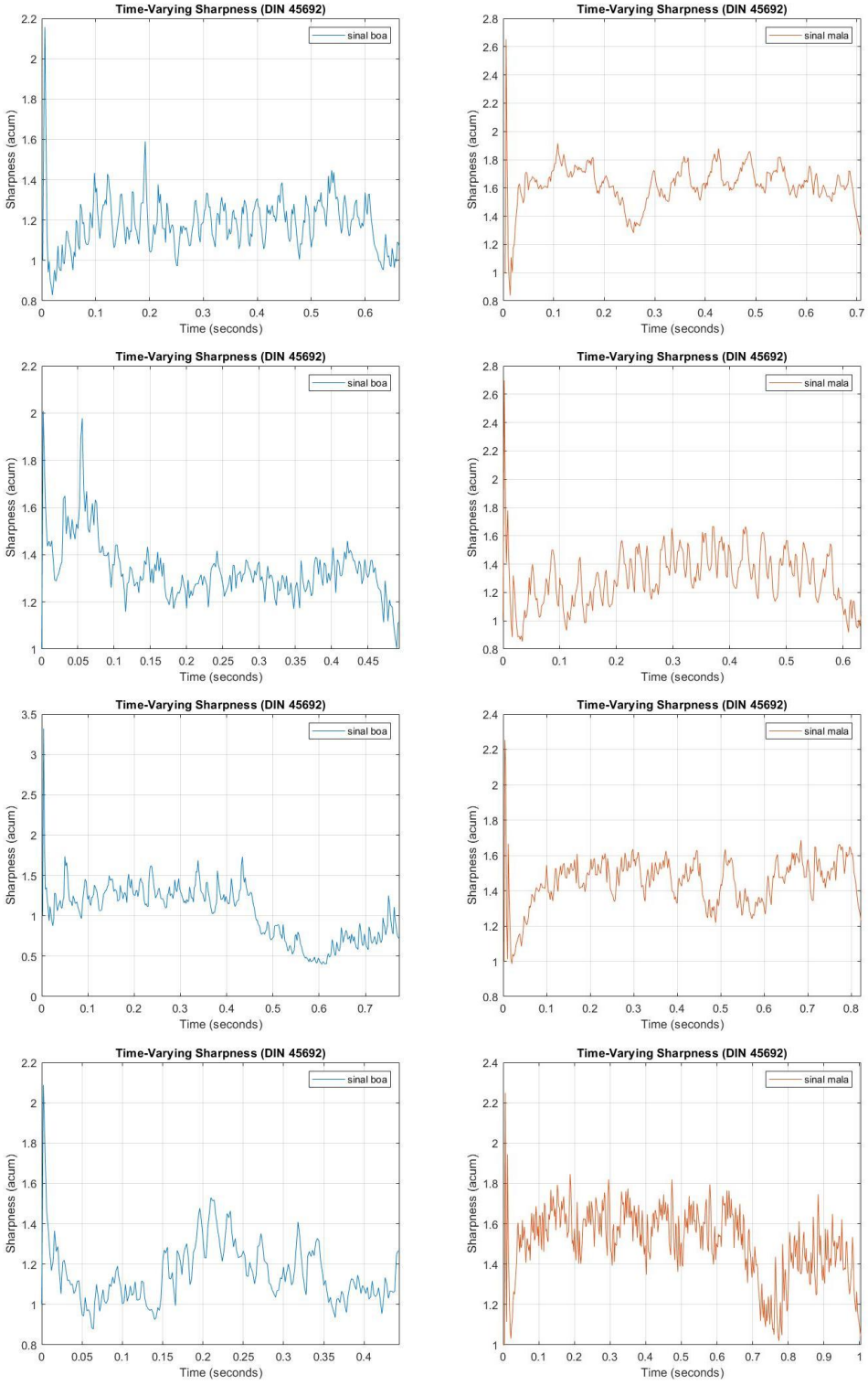
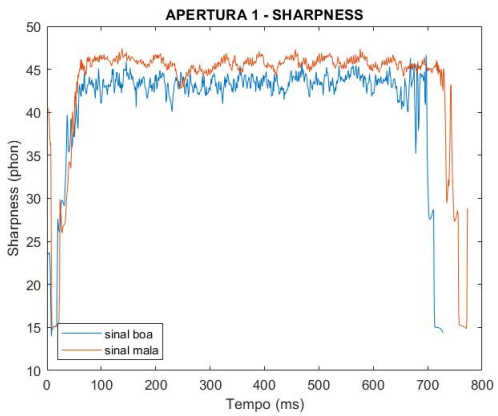
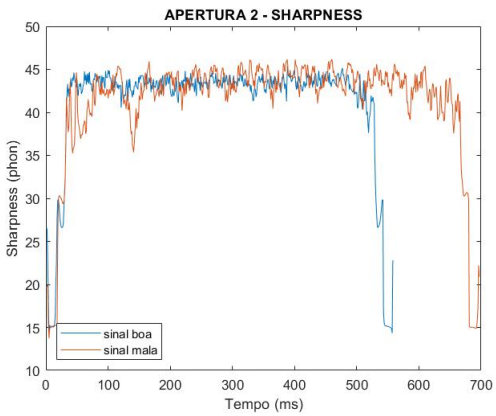


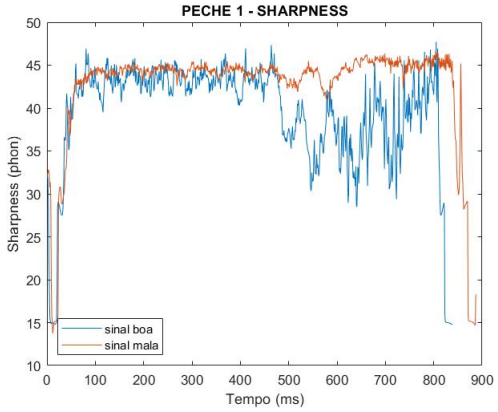
Figura 14: Resultados do *Sharpness* en *acum* na análise EOL



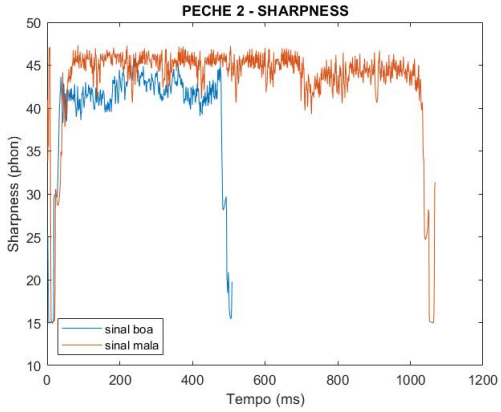
(a)



(b)

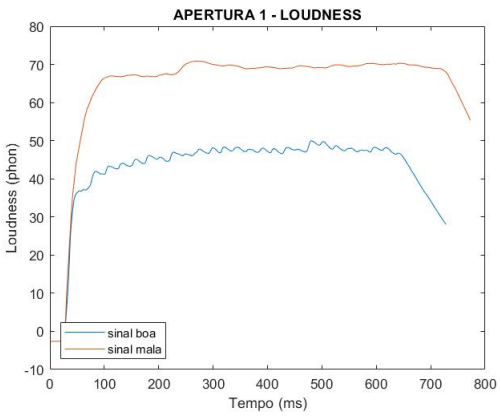


(c)

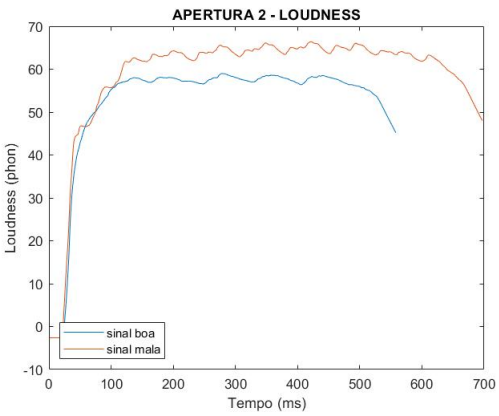


(d)

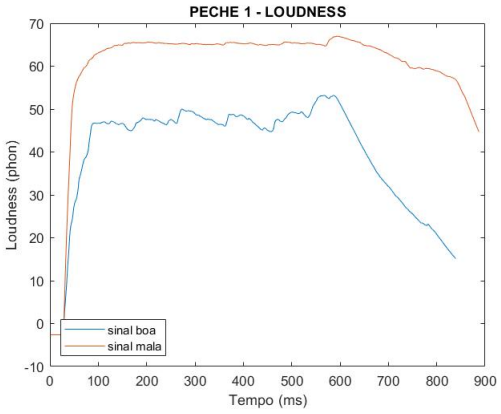
Figura 15: Resultados do *Sharpness* en *phon/ms* na análise EOL



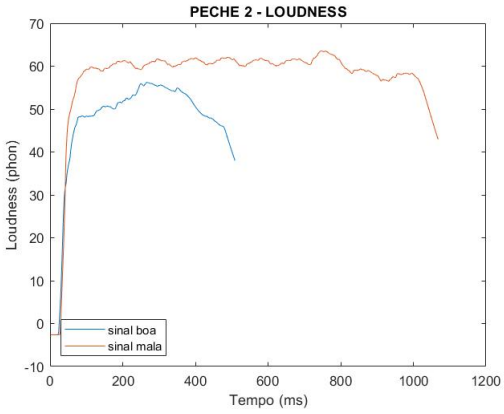
(a)



(b)



(c)



(d)

Figura 16: Resultados do *Loudness* en *phon/ms* na análise EOL

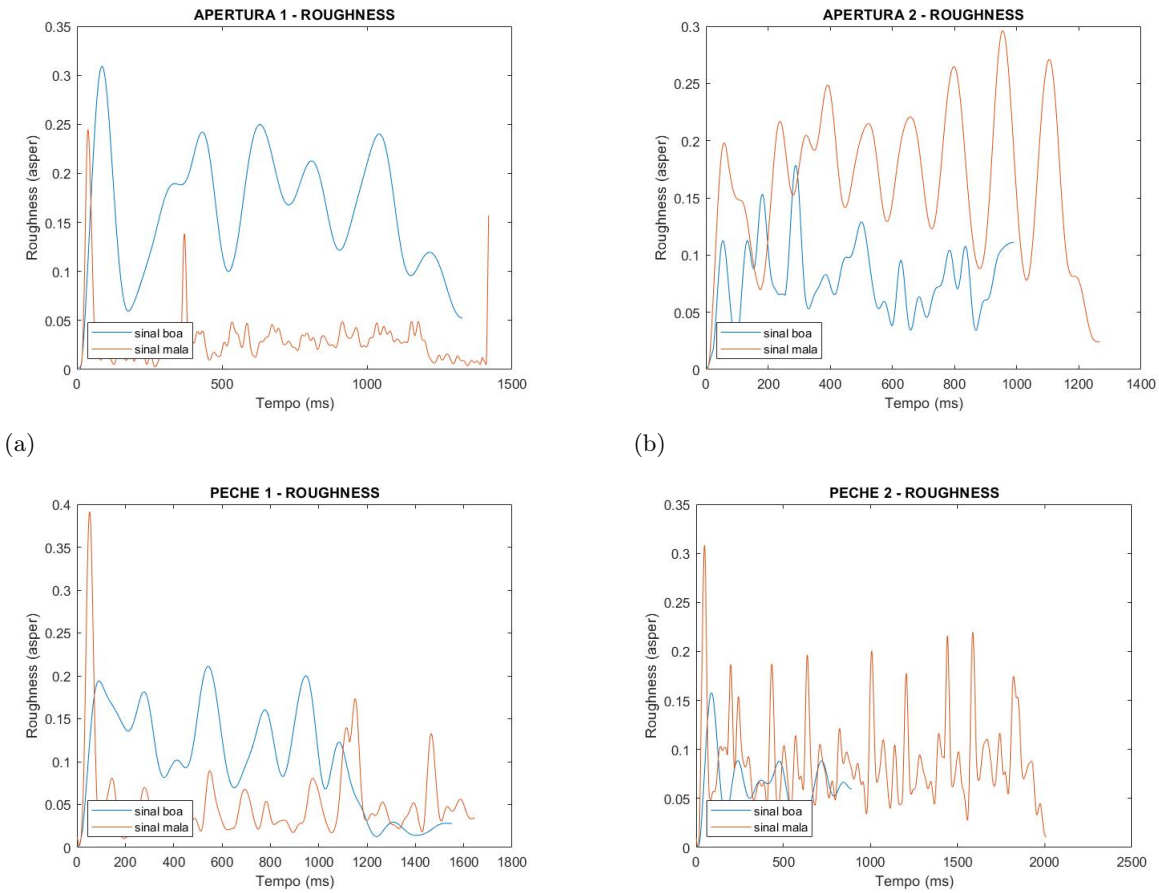
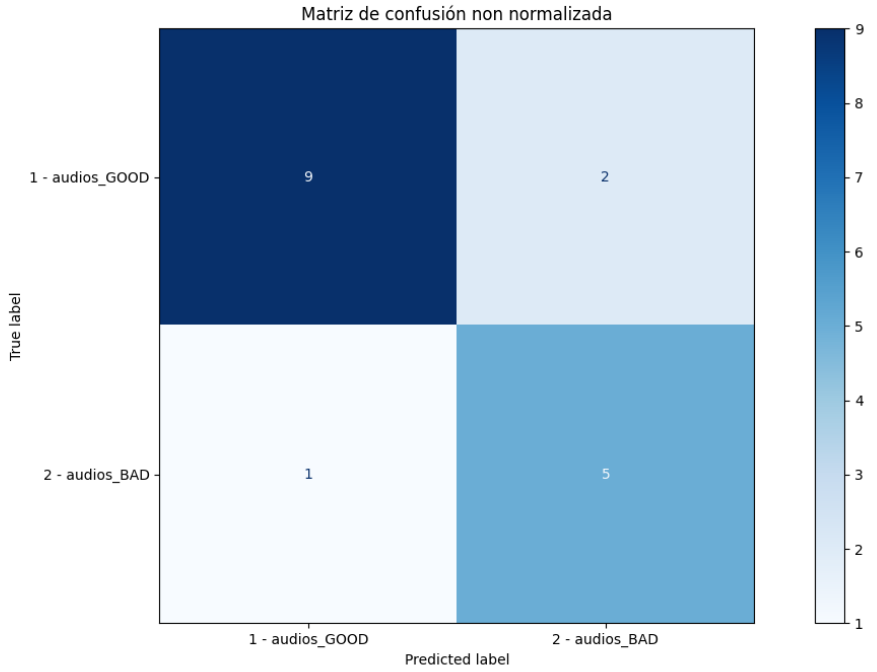
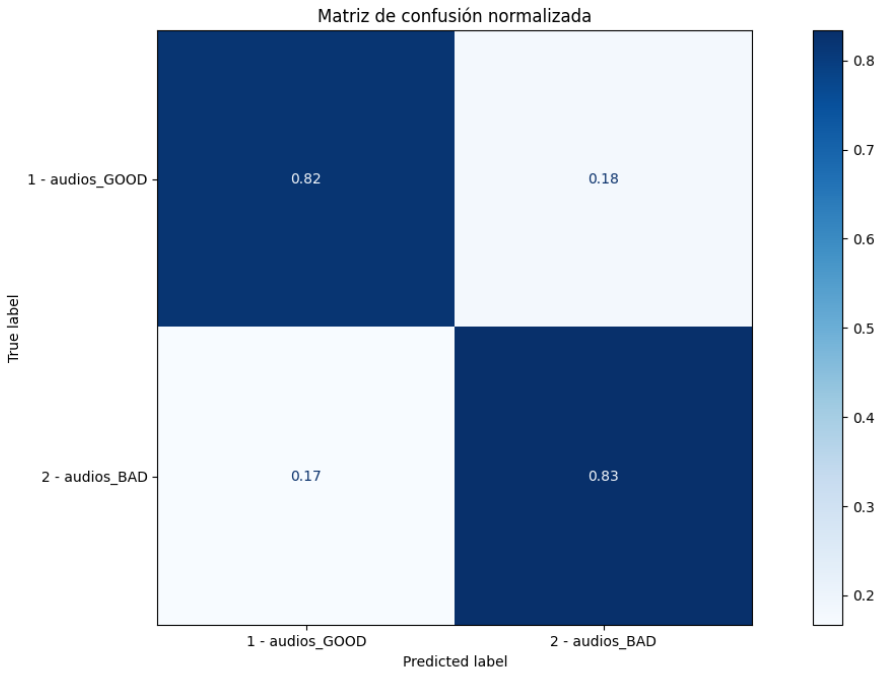


Figura 17: Resultados do *Roughness* en *asper* na análise EOL

B.2. Matrices de confusión

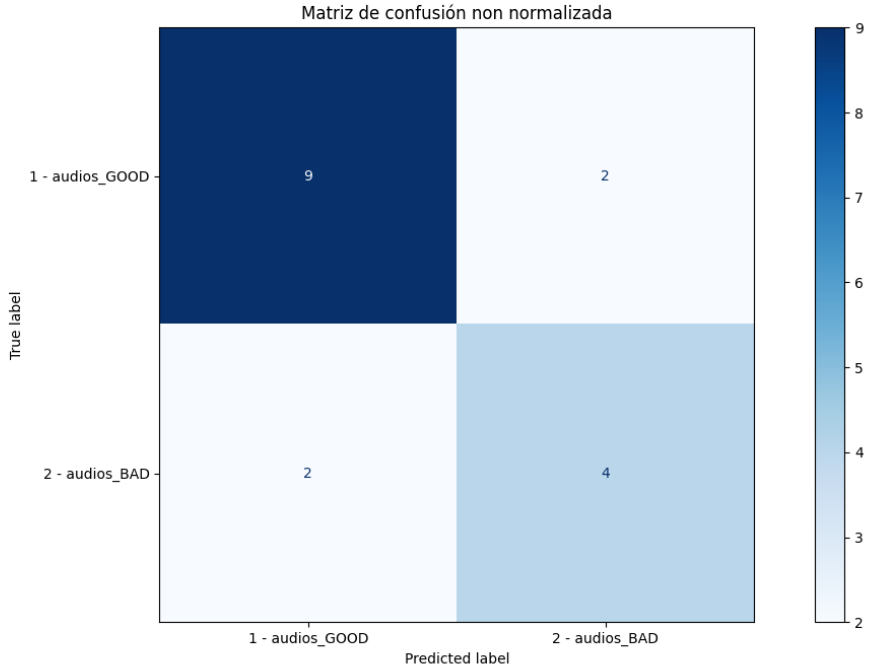


(a)

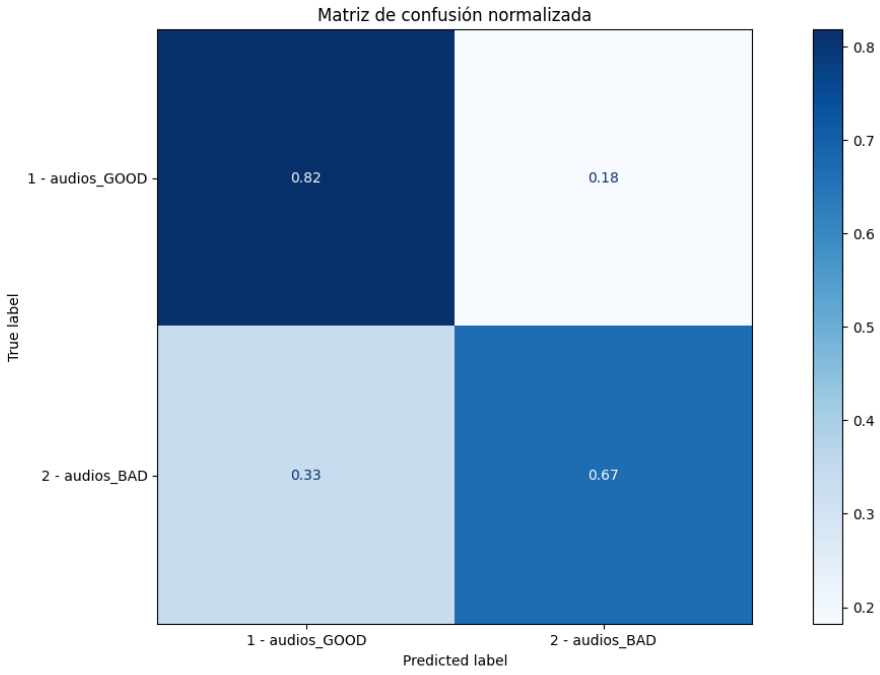


(b)

Figura 18: Matrices de confusión dos resultados de clasificar mediante *Loudness* (*a*: sen normalizar e *b* normalizada)

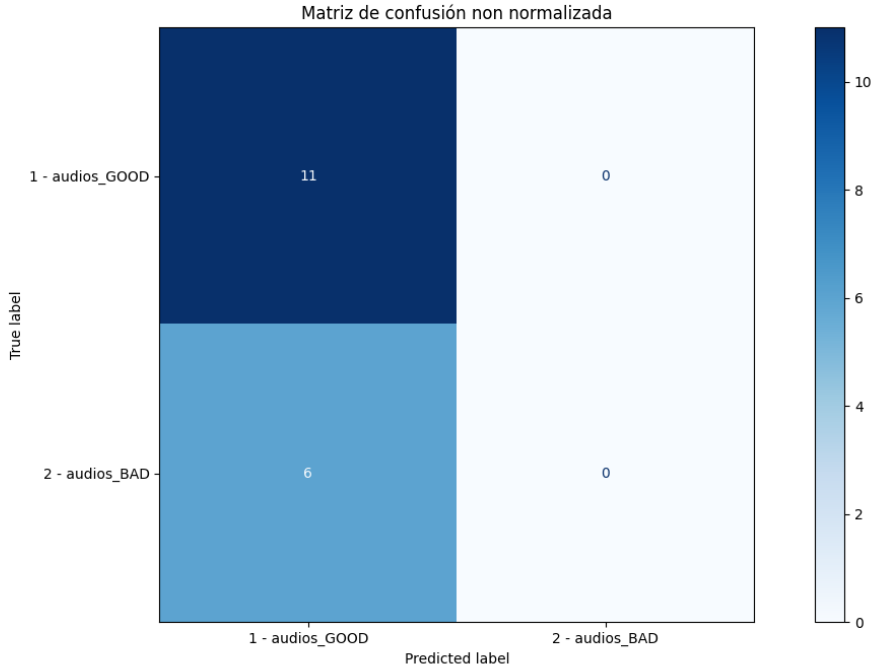


(a)

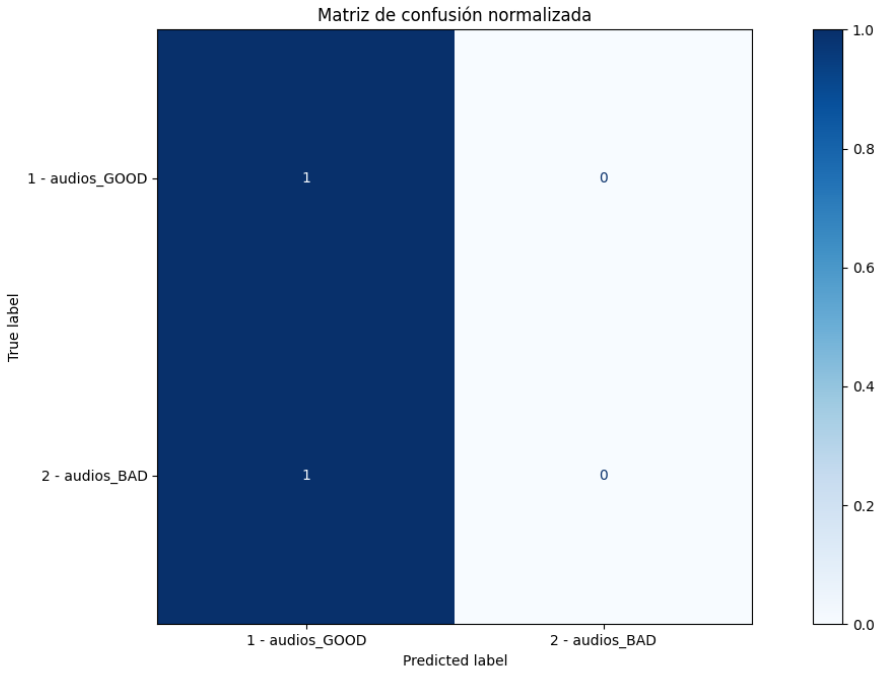


(b)

Figura 19: Matrices de confusión dos resultados de clasificar mediante *Sharpness* (*a*: sen normalizar e *b* normalizada)

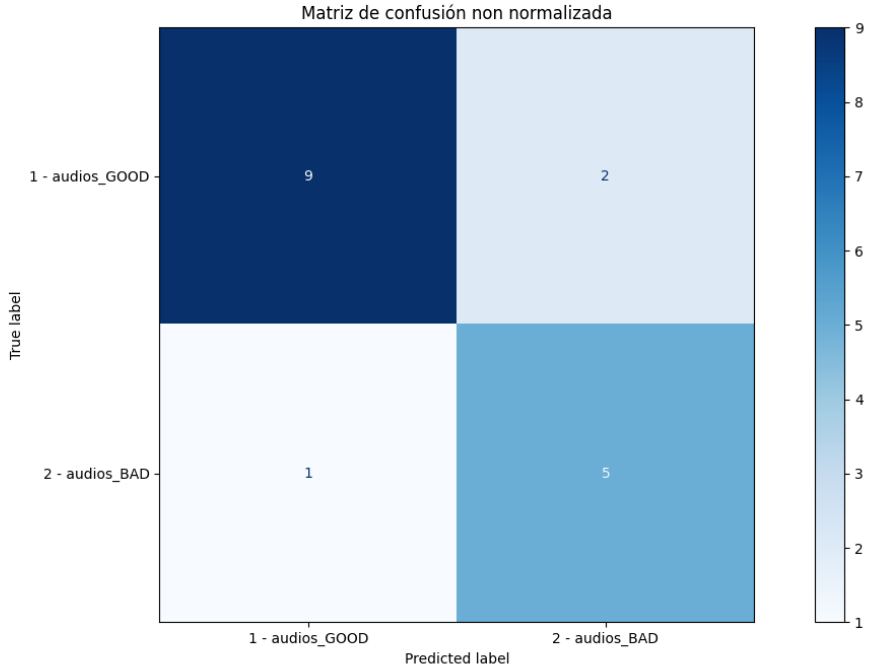


(a)

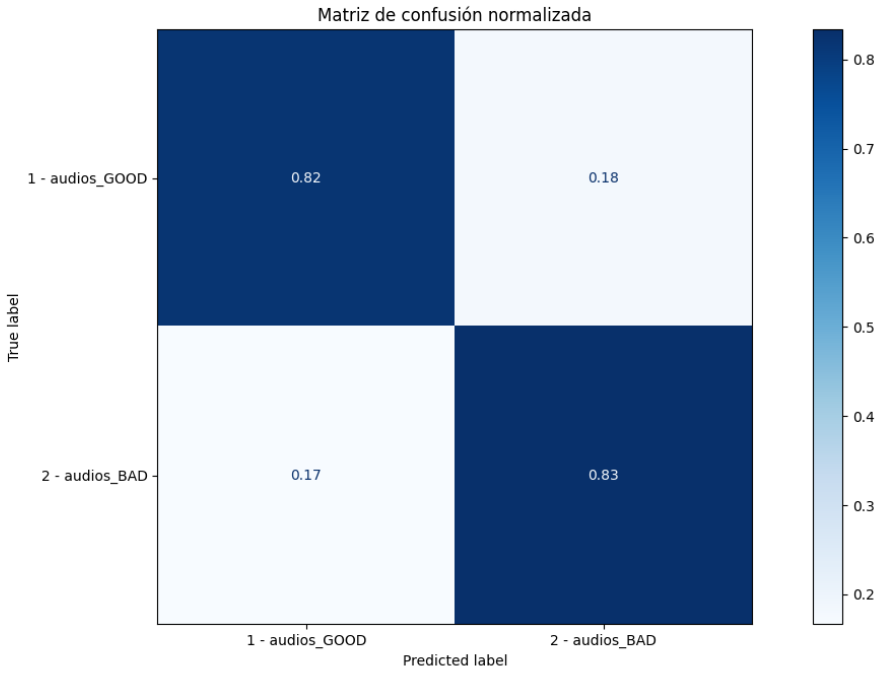


(b)

Figura 20: Matrices de confusión dos resultados de clasificar mediante *Roughness* (*a*: sen normalizar e *b* normalizada)

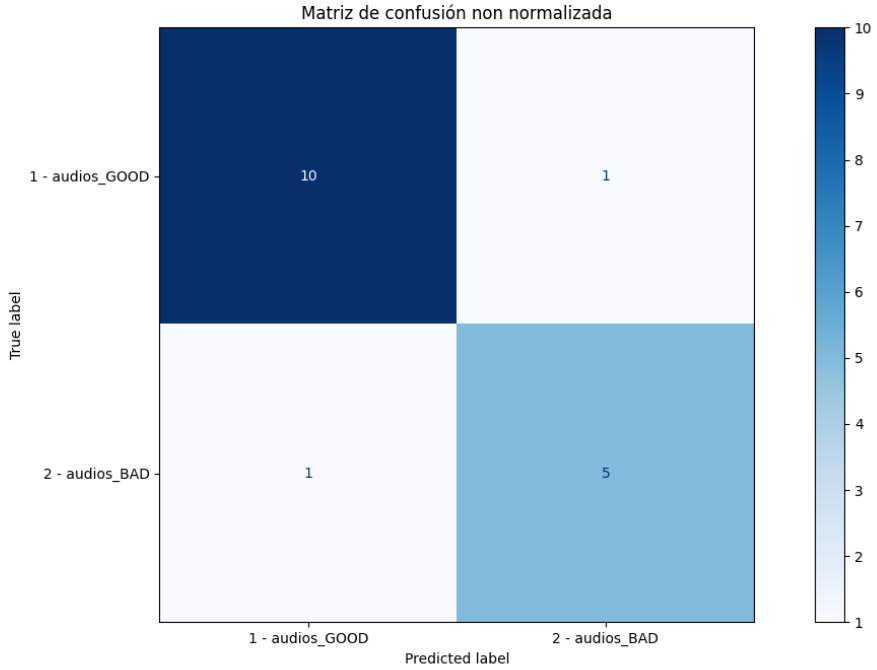


(a)

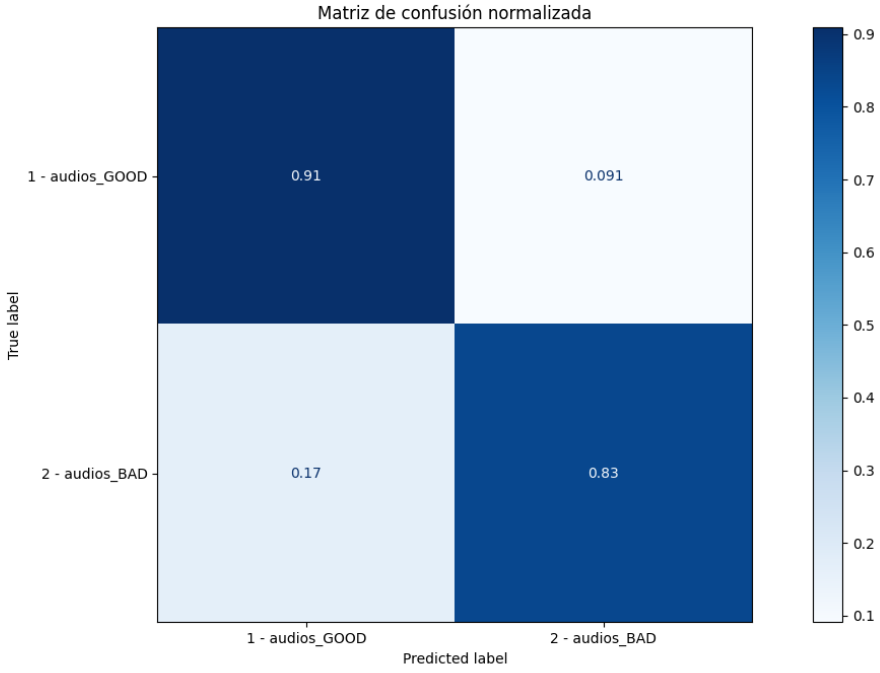


(b)

Figura 21: Matrices de confusión dos resultados de clasificar mediante *Loudness* e *Sharpness* (*a*: sen normalizar e *b* normalizada)

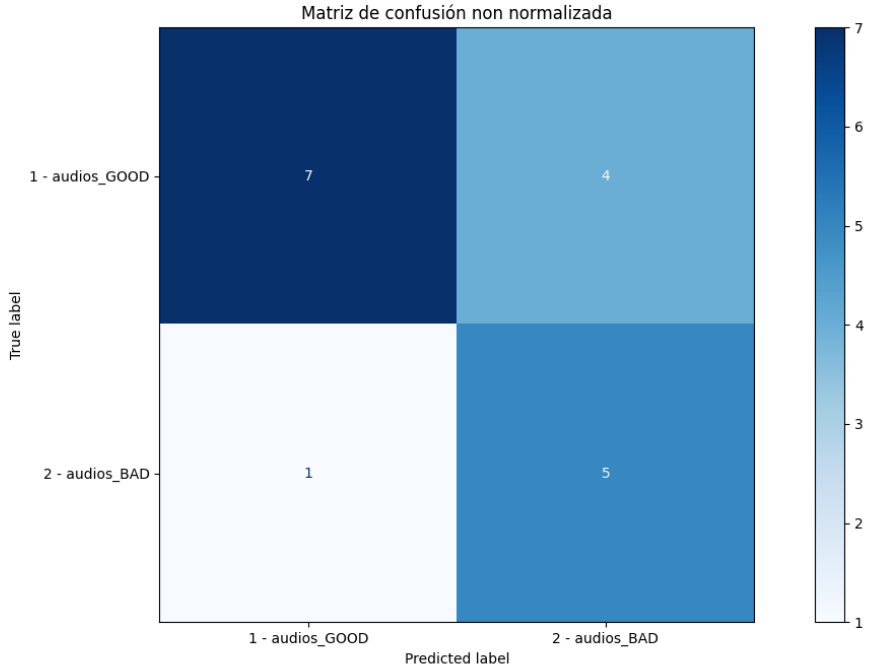


(a)

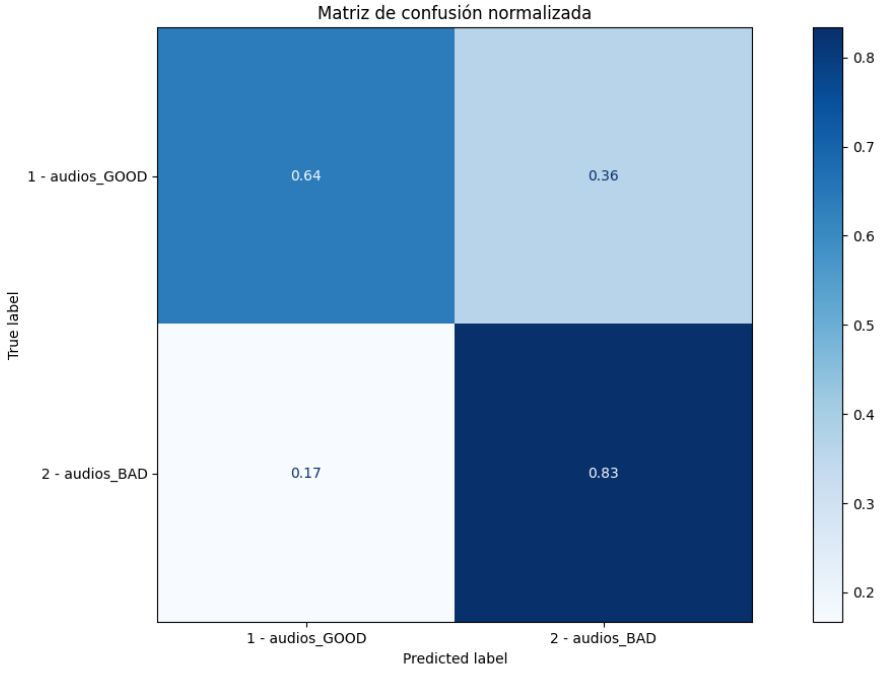


(b)

Figura 22: Matrices de confusión dos resultados de clasificar mediante *Loudness* e *Roughness* (*a*: sen normalizar e *b* normalizada)

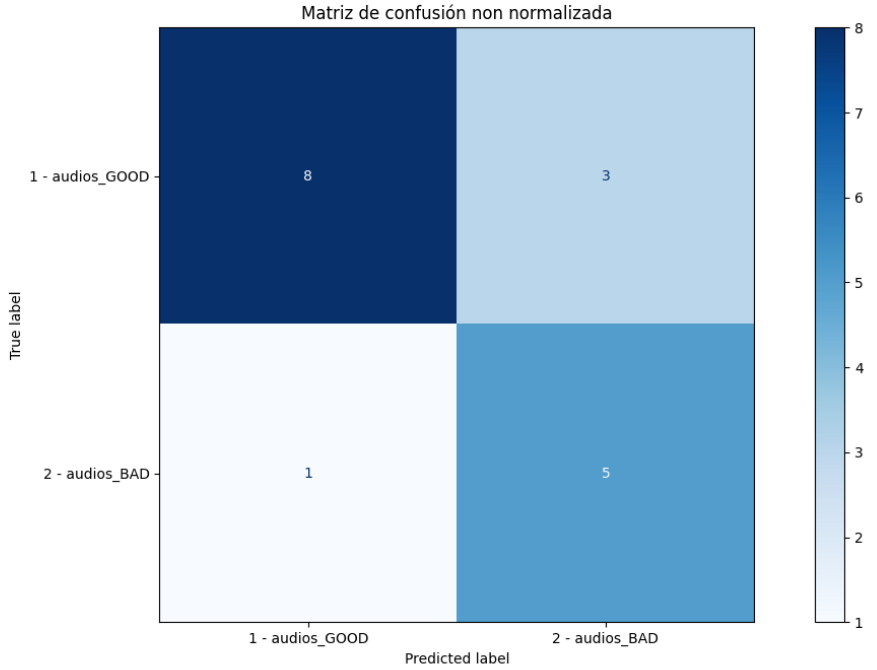


(a)

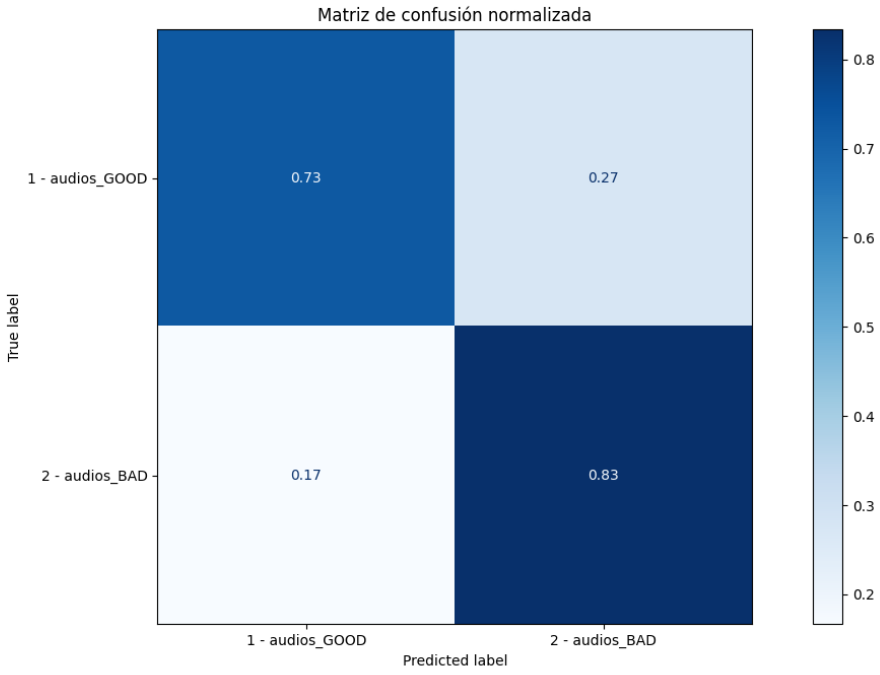


(b)

Figura 23: Matrices de confusión dos resultados de clasificar mediante *Sharpness* e *Roughness* (*a*: sen normalizar e *b* normalizada)



(a)



(b)

Figura 24: Matrices de confusión dos resultados de clasificar mediante *Loudness*, *Sharpness* e *Roughness* (*a*: sen normalizar e *b* normalizada)

Referencias

[1] Itu.int. n.d. [online] Available at: <https://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.1770-4-201510-I!!PDF-E.pdf>

[2] DIN (2009). DIN 45692-2009. Measurement Technique for the Simulation of the Auditory Sensation of Sharpness (Deutsches Institut fur Normung, Berlin)

[3] Dl.acm.org. Detection of Acoustic Events by using MFCC and Spectro-Temporal Gabor Filterbank Features | Proceedings of the 8th International Conference on Signal Processing Systems. [online] Available at: <https://dl.acm.org/doi/10.1145/3015166.3015186>

[4] REVISTA DE ROBOTS. 2021. Qué es la Industria 4.0 en 2020, la Cuarta Revolución Industrial. [online] Available at: <https://revistaderobots.com/industria/industria-4-0/>

[5] Daniel, P., and Weber, R. (1997). "Psychoacoustical Roughness: Implementation of an Optimized Model", Acta Acustica, Vol. 83: 113-123 https://www.microflown.com/markets/end-of-line-control

[6] Breebaart, J., & McKinney, M. F. (2004). Features for audio classification. InAlgorithms in Ambient Intelligence (pp. 113-129). Springer, Dordrecht.

[7] McFee, B. et al., 2015. librosa: Audio and music signal analysis in python. In Proceedings of the 14th python in science conference.

[8] J. Steinmetz, C. and D. Reiss, J., n.d. csteinmetz1/pyloudnorm. [online] GitHub. Available at: <https://github.com/csteinmetz1/pyloudnorm>

[9] Pearce, A., Safavi, S., Brookes, T., Mason, R., Wang, W. and Plumbley, M., 2019. Audio Commons: An Ecosystem for Creative Reuse of Audio Content. [online] Available at: <https://github.com/AudioCommons/timbral_models/tree/master/timbral_models>

[10] PyPI. 2021. mosqito. [online] Available at: <https://pypi.org/project/mosqito/>

[11] Pedregosa, F. et al., 2011. Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), pp.2825–2830.

[12] Harris, C.R. et al., 2020. Array programming with NumPy. Nature, 585, pp.357–362.

[13] Vuegen, Lode, et al. An MFCC GMM Approach for Event Detection and Classification. IEEE, 2013.

[14] Hugo Fastl and Eberhard Zwicker. 2006. Psychoacoustics: Facts and Models. Springer-Verlag, Berlin, Heidelberg.

[15] Havelok, D., Sonoko, K., Vorlander, M.: Handbook of Signal Processing in Acoustics. Springer, New York (2008)

[16] P. Daniel and R. Weber, "Psychoacoustical roughness: implementation of an optimized model,"

[17] Trends in audio signal feature extraction methods link: https://doi.org/10.1016/J.APACOUST.2019.107020 Authors Sharma, GarimaUmapathy, KartikeyanKrishnan, Sridhar

[18] GitHub. n.d. giulbia/baby_cry_detection. [online] Available at: <https://github.com/giulbia/baby_cry_detection>

[19] 1library.co. n.d. Sharpness. “ Nitidez, agudeza”, en acum .von Bismarck. [online] Available at: <https://1library.co/document/oz1r778q-sharpness-itidez-agudeza-en-acum-von-bismarck.html>