

Informe

Azahara Martínez, María de los Ángeles Díaz, Álvaro Nieva, Iyán Álvarez,
Florence Pellegrini, Óscar Camacho

2026-01-14

OBJETIVOS DEL PROYECTO

El objetivo principal de este trabajo es analizar señales de voz a partir de grabaciones propias y desarrollar modelos de clasificación capaces de identificar distintas características de la señal de audio. Para ello, se parte de un conjunto de audios etiquetados y se emplean técnicas de preprocesado, extracción de características y aprendizaje automático, con el fin de obtener modelos que puedan generalizar su comportamiento ante audios no utilizados durante el entrenamiento.

De manera más específica, los objetivos del trabajo son los siguientes:

- Analizar señales de voz grabadas por los propios autores y prepararlas para su posterior tratamiento mediante técnicas de preprocesado.
- Extraer características acústicas relevantes que permitan describir distintas propiedades de la voz de forma numérica.
- Construir un conjunto de datos estructurado a partir de las características extraídas y las etiquetas conocidas de cada audio.
- Desarrollar y entrenar modelos de clasificación capaces de identificar el sexo del hablante a partir de la señal de voz.
- Desarrollar modelos de clasificación para la identificación del acento, distinguiendo entre español neutro, andaluz y argentino.
- Investigar la viabilidad de diferenciar entre voces humanas y voces generadas por inteligencia artificial a partir de descriptores acústicos.
- Evaluar el rendimiento de los modelos propuestos mediante técnicas de validación adecuadas.

PREPROCESADO

Conversión y lectura de audios

Como primer paso del preprocesado, se desarrolló una función para la conversión de archivos de audio desde el formato `.m4a` al formato `.wav`, con el objetivo de unificar el tipo de señal y facilitar su posterior análisis. Para ello, se emplearon las librerías `av` y `tuneR` de R, que permiten la conversión y lectura de señales de audio de forma eficiente.

La función implementada convierte cada archivo a una señal monofónica con una frecuencia de muestreo de 16 kHz y guarda los archivos resultantes en un directorio específico. Además, se automatizó el proceso para convertir de manera recursiva todos los archivos `.m4a` contenidos en una carpeta y sus subdirectorios.

Detección y eliminación de ruido y silencios

Con el objetivo de mejorar la calidad de la base de datos y eliminar segmentos de audio que no aporten información, se implementó un algoritmo para discriminar entre fragmentos correspondientes al habla y aquellos correspondientes a ruidos o silencios, eliminando estos últimos a partir de un análisis de la *Short Time Energy* (STE) y el *Zero Crossing Rate* (ZCR). Este procedimiento genera nuevos archivos de audio que contienen únicamente la información acústica relevante.

El procesamiento se realizó mediante la creación de ventanas temporales, dividiendo la señal de entrada en tramos de 20 ms con un solapamiento del 50% entre ventanas consecutivas. Estos parámetros se seleccionaron para garantizar la cuasi-estacionariedad de la señal de voz, permitiendo un análisis espectral y temporal preciso sin perder continuidad en los bordes de cada tramo.

Para cada ventana se calcularon los dos descriptores mencionados anteriormente empleando **seewave** y **tuneR**:

- **Short Time Energy (STE):** Se calculó como la suma de los cuadrados de la amplitud de la señal en cada ventana, normalizando en el intervalo $[0, 1]$. Esta métrica actúa como el discriminador principal, asumiendo que los segmentos de voz presentan una energía significativamente superior frente a los ruidos de fondo o silencios.
- **Zero Crossing Rate (ZCR):** Se computó la frecuencia con la que la señal cambia de signo dentro de la ventana. Aunque el código permite su uso como criterio secundario para filtrar ruidos con alta frecuencia, en la configuración final se priorizó el criterio del STE.

El criterio de decisión se estableció mediante un umbral de energía definido. Aquellos tramos con una energía normalizada inferior a 0.02 (2% del máximo) se etiquetaron como *Ruido/Silencio*, mientras que las superiores se clasificaron como *Voz/Señal*. Finalmente, se reconstruyó la señal de audio concatenando temporalmente las muestras correspondientes a los segmentos válidos, descartando los tramos de silencio y exportando el resultado limpio a la carpeta de destino para su posterior caracterización.

EXTRACCIÓN DE CARACTERÍSTICAS

Las señales de voz contienen mucha información, pero trabajar directamente con el audio no resulta práctico para entrenar modelos de clasificación. Por ello, es necesario extraer una serie de características numéricas que describan la señal de forma más simple y manejable.

La extracción de características permite resumir aspectos importantes de la voz, como su tono, su energía o cómo se distribuye el sonido en distintas frecuencias. De esta manera, cada audio puede representarse mediante un conjunto de valores que capturan sus propiedades principales, evitando depender de la duración del archivo o de pequeñas variaciones que no aportan información relevante.

Estas características sirven como base para entrenar los modelos utilizados en el proyecto, ya que facilitan la comparación entre distintas voces y permiten identificar patrones asociados a diferentes tipos de hablantes o señales. Gracias a este proceso, es posible abordar tareas de clasificación como la identificación del sexo, el acento o la distinción entre voces humanas y generadas por inteligencia artificial.

Period pitch

El *period pitch* está directamente relacionado con la frecuencia fundamental de la voz, que corresponde a la vibración periódica más baja producida por las cuerdas vocales y constituye la base sobre la que se construyen el resto de componentes armónicos del habla. La frecuencia fundamental es especialmente relevante porque representa una característica estable de la voz y está estrechamente vinculada a la percepción de si una voz suena más grave o más aguda.

El period pitch se estimó mediante un método basado en la autocorrelación de la señal, que permite detectar la periodicidad dominante en los segmentos sonoros del audio, principalmente en vocales. A partir del retardo correspondiente al máximo de la autocorrelación se obtiene el período fundamental de la señal, y su inversa proporciona una estimación de la frecuencia fundamental o pitch, expresada en hercios (Hz), es decir, vibraciones por segundo.

De acuerdo con la literatura consultada, los valores típicos de frecuencia fundamental en voz humana se sitúan aproximadamente en el rango comprendido entre 85 y 260 Hz, presentando diferencias claras entre hombres y mujeres (Rabiner, 1978 y Basu, 2020).

A partir de estos rangos reportados, se definieron umbrales prácticos para la clasificación del sexo. Con el fin de evitar clasificaciones forzadas en las zonas de solapamiento entre ambos grupos, se estableció una región intermedia de indeterminación. De este modo, se consideraron voces masculinas aquellas con frecuencias fundamentales inferiores a 170 Hz, y voces femeninas aquellas con valores superiores a 180Hz.

Para calcular el pitch creamos la función `period_pitch` y, con el fin de garantizar la fiabilidad de la estimación, se incorporan una serie de comprobaciones previas sobre la señal de entrada. En primer lugar, se verifica que el audio contenga un número suficiente de muestras, ya que señales excesivamente cortas no permiten identificar una periodicidad clara. Asimismo, la búsqueda del período fundamental se restringe a un rango de retardos coherente con los valores esperados de frecuencia fundamental en voz humana, evitando así detecciones erróneas asociadas a componentes no relevantes de la señal. En aquellos casos en los que la longitud efectiva de la señal no permite cubrir dicho rango de retardos, el cálculo del pitch se considera no válido.

Esta decisión se basa en el hecho de que los métodos basados en autocorrelación solo proporcionan estimaciones fiables en segmentos sonoros suficientemente largos y con una periodicidad bien definida, como ocurre principalmente en las partes vocales del habla.

Coeficientes Mel (MFCC) y representación de la señal

Como parte de la extracción de características espectrales, se calcularon los coeficientes cepstrales en escala Mel (MFCC), ampliamente utilizados en tareas de análisis de voz por su capacidad de capturar información relacionada con el timbre y la envolvente espectral. Para ello, se aplicó el procedimiento estándar por ventanas cortas: se dividió cada audio en tramas solapadas (ventana de 25 ms y salto de 10 ms), se realizó la transformada rápida de Fourier (FFT) por trama y se obtuvo el espectro de potencia. Posteriormente, dicho espectro se proyectó sobre un banco de filtros en escala Mel (40 bandas), se aplicó una compresión logarítmica y, finalmente, una transformada discreta del coseno (DCT) para obtener un conjunto compacto de coeficientes (12 MFCC por trama). Adicionalmente, se aplicó un pre-énfasis para compensar la caída de energía en altas frecuencias típica de señales de voz.

Con el fin de incorporar información dinámica, se calcularon también las derivadas temporales de primer orden (Δ) y segundo orden ($\Delta\Delta$), que aproximan respectivamente la *velocidad* y la *aceleración* de los MFCC a lo largo del tiempo.

Para adaptar esta representación por tramas a modelos clásicos de aprendizaje automático, cada audio se transformó en un único vector de características agregando estadísticas globales: media y desviación típica de los MFCC, así como de sus derivadas Δ y $\Delta\Delta$. De este modo, cada clip queda representado por un vector fijo que resume tanto el contenido espectral promedio como su variabilidad temporal.

Centroide espectral

Centroide espectral global El centroide espectral global se calculó como una medida resumen de la distribución energética del espectro de cada señal de audio. Este descriptor representa la frecuencia media ponderada por la potencia espectral y está relacionado con la percepción de brillo del sonido. Para su cálculo, se eliminó previamente el componente de continua de la señal con el fin de evitar un sesgo artificial hacia bajas frecuencias.

El análisis del centroide espectral permitió observar diferencias en los valores medios entre voces humanas y voces generadas por inteligencia artificial, especialmente al separar los resultados por sexo. No obstante, aunque este descriptor mostró cierto potencial discriminativo, su uso de forma aislada resultó limitado debido al solapamiento existente entre clases.

```
spectral_centroid <- function(w) {
  fs <- as.numeric(w@samp.rate)
  x <- as.numeric(w@left)

  # quitar media para estabilidad
  x <- x - mean(x)

  N <- length(x)

  X <- fft(x)
  K <- floor(N / 2) + 1

  P <- Mod(X[1:K])^2 #potencia espectral
  denom <- sum(P) #denominador del centroide
  if (!is.finite(denom) || denom <= 0) return(0)

  # Evitar overflow: usar double desde el principio
  freqs <- as.numeric(0:(K - 1)) * (fs / as.numeric(N))

  sum(freqs * P) / denom
}
```

Segmentación temporal en 12 partes Con el objetivo de analizar la evolución temporal del centroide espectral y evaluar la estabilidad espectral de las señales, se propuso una segmentación temporal de cada audio en doce partes consecutivas de igual duración.

Para cada segmento se calculó de forma independiente el centroide espectral, obteniendo así un vector de doce valores por señal. Esta aproximación permite capturar variaciones temporales del contenido espectral que no son visibles al emplear un único centroide global. A partir de estos valores se derivaron medidas estadísticas como la desviación típica y las diferencias entre centroides consecutivos, con el fin de caracterizar la variabilidad espectral de cada voz.

Aunque estos descriptores no mostraron una separación clara entre voces humanas y sintéticas cuando se analizaron de forma individual, su combinación permitió capturar patrones más complejos que resultaron útiles en modelos de clasificación no lineales.

```
spectral_centroid_12 <- function(w, n_parts = 12) {
  fs <- as.numeric(w@samp.rate)
  x <- as.numeric(w@left)
  x <- x - mean(x) #evita que se sesgue el centroide a frecuencias bajas

  N <- length(x)
  if (N < n_parts * 10 || fs <= 0) return(rep(NA_real_, n_parts)) #por si el audio es demasiado corto

  seg_len <- floor(N / n_parts) #número de muestras por segmento
  cents <- rep(NA_real_, n_parts)

  for (i in seq_len(n_parts)) {
    start <- (i - 1) * seg_len + 1
```

```

end   <- if (i < n_parts) i * seg_len else N
seg  <- x[start:end]
seg  <- seg - mean(seg)

L <- length(seg)
if (L < 2) next #por si el segmento es demasiado corto

X <- fft(seg)
K <- floor(L / 2) + 1
P <- Mod(X[1:K])^2
denom <- sum(P)
if (!is.finite(denom) || denom <= 0) next

freqs <- as.numeric(0:(K - 1)) * (fs / as.numeric(L))
cents[i] <- sum(freqs * P) / denom
}

cents
}

```

CONSTRUCCIÓN DEL CONJUNTO DE DATOS

Para organizar y analizar los 466 audios, se creó un dataframe que contiene las características y etiquetas de cada audio, donde denotamos:

- M: masculino / F: femenino
- P: persona / IA: IA
- AN: andaluz / AR: argentino / N: neutro

Primero, se creó un dataframe que etiquetara cada persona con su sexo, origen y acento, mediante la función `etiquetar_por_nombre()`, que compara el nombre del archivo con los nombres del dataframe de reglas y devuelve las etiquetas correspondientes.

A continuación, para cada audio se incluyeron las siguientes variables: duración, zcr, energía rms y pitch. Para ello se utilizó la función `añadir_voz()` la cuál recibe un archivo de audio, extrae sus características, asigna las etiquetas usando la función anterior y añade toda la información como una nueva fila al dataframe. Así, cada fila del dataframe corresponde a un audio individual, con sus características numéricas y etiquetas.

Cabe destacar que todas las características consideradas en el conjunto de datos están definidas como medias o medidas normalizadas respecto a la longitud de la señal, de modo que no dependen de la duración total del audio. En consecuencia, el proceso de eliminación de silencios únicamente reduce las regiones no informativas de la señal, sin afectar a la coherencia ni a la comparabilidad de las características extraídas entre distintos audios.

Para automatizar la creación del dataframe, se implementó la función `df_carpeta()`, que recorre todos los archivos de la carpeta con los audios limpios y aplica `añadir_voz()` a cada uno. De forma que, se obtiene un dataframe completo, `voces_df`, con toda la información lista para análisis y clasificación.

Construcción del conjunto de datos para detección de I: MFCC

Para cada archivo de audio se aplicó un preprocesado común con el objetivo de asegurar la comparabilidad entre señales. En concreto, se convirtió cada audio a mono mediante la función `to_mono()`, promediando los

canales izquierdo y derecho en caso de que la señal fuese estéreo. Después, se homogeneizó la frecuencia de muestreo a un valor fijo (16 kHz) mediante la función `resample_if_needed()`, ya que el cálculo de MFCC depende directamente de la escala de frecuencias y no es comparable si los audios están a distintas tasas de muestreo.

Una vez normalizada la señal, se extrajeron los coeficientes MFCC utilizando una configuración fija de ventanas temporales (duración de ventana y salto constantes) y un número determinado de coeficientes cepstrales, para ello se usó la función `mefcc()` de la librería `tuner`. Dado que los MFCC se calculan por frames y generan una matriz temporal, se incorporó además información dinámica calculando las derivadas temporales: (Δ) (delta) y $(\Delta\Delta)$ (delta-delta), mediante la función `delta_simple()`, capturando así los cambios espectrales a lo largo del tiempo.

Para transformar estas matrices (MFCC, (Δ) y $(\Delta\Delta)$) en un conjunto de variables de tamaño fijo por observación, se resumió cada una mediante estadísticas agregadas. En particular, para cada coeficiente se calcularon su media y su desviación estándar a lo largo de todos los frames del audio. Además, se incluyó un control de calidad eliminando valores no finitos (NaN o infinitos) y descartando los audios excesivamente cortos, con el fin de evitar que la extracción de características generase vectores inestables o no representativos.

Para incorporar toda esta información en el conjunto de datos, se utilizó la función `extract_feat_from_wav()`, la cual recibe un archivo de audio, realiza el preprocesado descrito, calcula MFCC, (Δ) y $(\Delta\Delta)$, obtiene los estadísticos (medias y desviaciones), asigna la etiqueta correspondiente y devuelve un vector final de características. Así, cada fila del dataframe corresponde a un audio individual, con sus variables MFCC resumidas y su clase asociada (real o IA), además de metadatos como la ruta del archivo y la frecuencia de muestreo final.

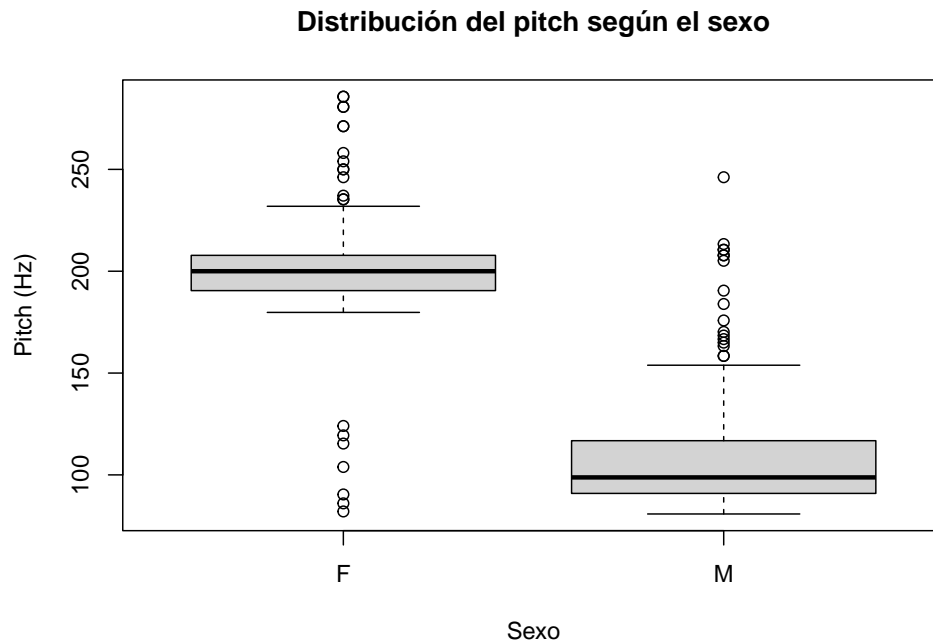
Para automatizar la creación del dataframe completo, se implementó la función `build_feat_dataframe()`, que recorre todos los archivos .wav de las carpetas real y fake, aplica `extract_feat_from_wav()` a cada uno y une los resultados en un único dataframe.

ANÁLISIS EXPLORATORIO

Antes de aplicar los métodos de clasificación, se realizó un análisis exploratorio del conjunto de datos con el objetivo de comprender la distribución de las variables y detectar posibles patrones relevantes. Así, el conjunto de datos final está compuesto por 466 audios, con una distribución equilibrada entre voces femeninas (240) y masculinas (226), lo que reduce la posibilidad de sesgos derivados del desbalanceo de clases.

Tras hacer un análisis descriptivo usando `summary()`, se puede ver que la duración de los audios después de limpiarlos y eliminar los silencios presenta una variabilidad moderada, con valores comprendidos aproximadamente entre 1.5 y 5 segundos. Por otro lado, las características ZCR y energía RMS presentan rangos consistentes y valores medios estables, indicando señales con contenido sonoro continuo y sin presencia dominante de ruido.

En cuanto a la frecuencia fundamental (pitch), se observa un rango amplio de valores (aproximadamente entre 80 y 285 Hz), consistente con voces humanas. El análisis por sexo revela que las voces femeninas presentan valores de pitch más elevados que las masculinas. Si lo representamos usando un diagrama de cajas, podemos ver un solapamiento entre ambas distribuciones, lo que indica que probablemente existan errores de clasificación en algunos audios concretos al emplear únicamente esta característica.



Por último, el análisis de correlación entre las variables numéricas muestra una alta correlación entre la duración y la ZCR, mientras que el pitch presenta una correlación prácticamente nula con el resto de variables. Esto sugiere que el pitch aporta información complementaria y relevante para la tarea de clasificación.

```
##          duracion  zcr energia_rms period_pitch
## duracion      1.00 0.99      0.20      0.01
## zcr           0.99 1.00      0.21      0.04
## energia_rms   0.20 0.21      1.00      0.00
## period_pitch  0.01 0.04      0.00      1.00
```

MODELOS DE CLASIFICACIÓN Y VALIDACIÓN

Clasificación del sexo

A partir del conjunto de características extraídas y del análisis exploratorio realizado, tenemos como objetivo la clasificación del sexo del hablante. De esta forma, se plantea el problema como una clasificación binaria y se emplea un modelo de regresión logística. Este tipo de modelo permite relacionar las características extraídas con la probabilidad de que una voz corresponda a un hablante masculino o femenino.

Para ello, se codificó la variable objetivo **sexo** en formato numérico, asignando el valor 1 a las voces femeninas y 0 a las masculinas. Esta transformación permite el uso de modelos probabilísticos basados en regresión logística.

Inicialmente, se ajustó un modelo de regresión logística multivariante que incluía todas las características disponibles en el conjunto de datos: duración, tasa de cruces por cero (ZCR), energía RMS, frecuencia fundamental (pitch), así como las variables categóricas de origen y acento. Este primer modelo tiene como objetivo evaluar la contribución conjunta de todas las variables y explorar posibles relaciones entre ellas y la variable respuesta.

Al analizar el resumen del modelo de regresión logística completo mediante la función `summary()`, se observó que la única variable estadísticamente significativa para la clasificación del sexo era la frecuencia fundamental (period pitch), mientras que el resto de variables no aportaban evidencia significativa una vez incorporado dicho descriptor. A partir de esta observación, se decidió contrastar un modelo reducido basado exclusivamente en el pitch frente al modelo completo mediante un ANOVA, con el fin de comprobar si la complejidad adicional estaba justificada.

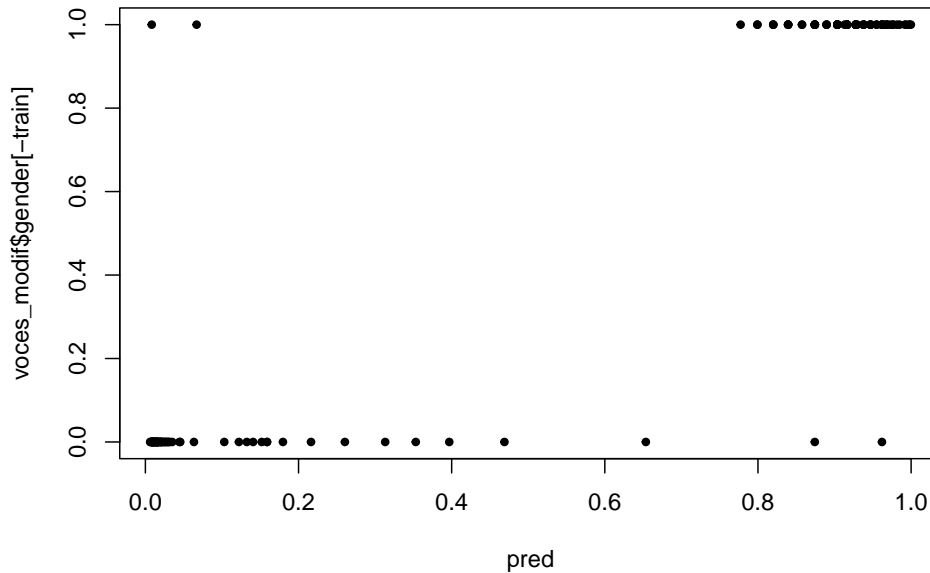
```
## Analysis of Deviance Table
##
## Model 1: gender ~ period_pitch
## Model 2: gender ~ sexo + origen + acento + duracion + zcr + energia_rms +
##          period_pitch
##      Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1          464      186.78
## 2          457         0.00  7   186.78 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

El contraste mediante ANOVA entre el modelo reducido y el modelo completo muestra que incorporar el resto de variables produce una reducción significativa de la devianza ($p < 0.001$). No obstante, el análisis individual de los coeficientes revela que únicamente el period pitch resulta estadísticamente significativo, mientras que el resto de variables no aportan información relevante de forma independiente.

A continuación, se evaluó el desempeño del modelo reducido en un esquema de entrenamiento y prueba. Para ello, se seleccionó aleatoriamente un 70% de las observaciones para ajustar el modelo y se reservó el 30% para validación.

La bondad de ajuste del modelo se reflejó en un pseudo- R^2 de 0.697, lo que sugiere que aproximadamente el 69.7% de la variabilidad del conjunto de entrenamiento queda explicada por el modelo basado únicamente en el period pitch. Posteriormente, se aplicó el modelo al conjunto de prueba para evaluar su capacidad predictiva.

La correlación entre las probabilidades predichas y las clases reales alcanzó un valor de 0.918, evidenciando una buena concordancia entre las predicciones y las etiquetas de sexo.



Al observar la gráfica de predicciones, se aprecia que la mayoría de los puntos correspondientes a voces masculinas se agrupan cerca de 0, mientras que las voces femeninas se concentran alrededor de 1, lo que indica que el modelo reconoce correctamente la mayoría de los casos. Sin embargo, algunos puntos se sitúan en valores intermedios, mostrando incertidumbre en la predicción.

Sin embargo, en el análisis exploratorio vimos que el diagrama de cajas del pitch tenía zonas donde había solapamiento entre los rangos de frecuencia fundamental de hombres y mujeres. Por lo tanto, es probable que los audios con pitch en el rango intermedio sean precisamente los que generan errores de clasificación, reflejando la variabilidad natural del tono de voz y características individuales que no se capturan únicamente con el period pitch.

En conjunto, estos resultados refuerzan que el pitch es el descriptor más relevante para la clasificación del sexo, pero que la superposición natural de frecuencias humanas introduce un margen de error inevitable, explicando los puntos intermedios en la gráfica de predicciones.

Clasificación del acento

Además de la clasificación por sexo, se abordó la identificación del acento de la voz, considerando tres categorías: español neutro, andaluz y argentino. Para esta tarea se empleó un modelo de regresión logística multinomial, adecuado para problemas de clasificación con más de dos clases.

Como variables predictoras se utilizaron descriptores acústicos de bajo nivel, concretamente la tasa de cruces por cero (ZCR), la energía RMS y la duración del audio. Estas características se seleccionaron por su relación con aspectos articulatorios y prosódicos que pueden variar entre acentos.

Con el fin de explorar la separabilidad entre clases, se realizó un análisis gráfico de la distribución del ZCR en función del acento, observándose diferencias apreciables entre algunas categorías. Posteriormente, se ajustó el modelo multinomial y se evaluó su capacidad explicativa mediante el pseudo- R^2 de McFadden, comparando la verosimilitud del modelo completo con la de un modelo nulo.

Aunque los resultados muestran que estas características contienen información relevante para la clasificación del acento, el rendimiento del modelo está condicionado por el tamaño reducido del conjunto de datos, por lo que se considera necesario ampliar el número de muestras para obtener conclusiones más robustas.

Clasificación del origen (IA vs Humano)

Para tratar este problema de clasificación, se probaron diferentes datasets y dataframes, con los que se entrenó un modelo *SVM con kernel RBF*.

Se eligió entrenar un *SVM con kernel RBF* porque es un modelo especialmente adecuado para el tipo de datos que hemos usado.

En nuestro caso, cada audio no se mantiene como una secuencia temporal completa, sino que se transforma en un vector fijo de características: medias y desviaciones estándar de los MFCC y de sus derivadas (Δ) y ($\Delta\Delta$). Esto convierte el problema en un dataset tabular (una fila por audio y un número moderado de variables, en torno a 72 predictores), similar a un problema clásico de clasificación con variables numéricas.

Con este tipo de variables, es razonable esperar que la separación entre audios reales y audios generados por IA no sea perfectamente lineal. Es decir, las dos clases pueden mezclarse en el espacio de características y necesitar una frontera de decisión más flexible que una simple recta o plano.

Por este motivo se utiliza un SVM con kernel RBF, ya que este modelo:

- Puede capturar relaciones no lineales entre las variables.
- Suele funcionar muy bien cuando hay un número medio de características (ni muy pocas ni miles)
- Es un baseline clásico y robusto en tareas de voz y audio cuando se trabaja con MFCC y estadísticas agregadas. (Temko and Nadeu 2006)

Proceso de entrenamiento

Para entrenar el clasificador se definió un procedimiento de validación y ajuste de hiperparámetros que permite estimar el rendimiento de forma robusta y reducir el riesgo de sobreajuste.

En primer lugar, se estableció un esquema de validación mediante la función `trainControl()`. En este caso se utilizó validación cruzada repetida (repeated cross-validation), dividiendo el conjunto de entrenamiento en 5 particiones (5 folds), de forma que en cada iteración se entrena el modelo con 4 folds y se valida con el fold restante. Además, el proceso completo se repite `repeats = 2` veces. Una vez fijado el esquema de validación, se entrenó el modelo mediante la función `train(method="svmRadial")` del paquete `caret`, obteniendo el modelo entrenado: `svm_fit`.

En total se entrenaron 2 modelos, con 2 datasets diferentes: un dataset extraído de internet y otro dataset formado por audios nuestros y generados con IA por nosotros.

Proceso de testeo

Para evaluar el rendimiento del modelo una vez entrenado, en primer lugar, se generaron las predicciones del modelo sobre el conjunto de test mediante la función `predict()`. Por un lado, se solicitaron las probabilidades por clase, extrayendo concretamente la probabilidad asociada a la clase “ia” y por otro lado, se obtuvo también la predicción final de clase.

Además, se calcularon métricas de clasificación a partir de estas predicciones. En primer lugar, se utilizó `confusionMatrix()` para construir la matriz de confusión comparando las clases predichas y se evaluó el modelo desde un punto de vista probabilístico mediante la curva ROC, empleando la función `roc()` del paquete `pROC`.

Se realizó el testeo sobre datos del dataset extraído de internet y sobre datos extraídos del dataset creado por nosotros.

Resultados

Se evaluó el rendimiento del sistema bajo tres configuraciones experimentales, combinando distintos conjuntos de entrenamiento y test con el objetivo de analizar tanto el desempeño en condiciones controladas como su capacidad de generalización:

1. Entrenamiento con el dataset de Internet y test con el dataset de Internet, para medir el rendimiento del modelo en el mismo dominio de datos con el que ha sido entrenado.
2. Entrenamiento con el dataset de Internet y test con nuestro dataset, para evaluar la capacidad de generalización del modelo al aplicarlo sobre audios obtenidos en un entorno distinto (cambio de dominio).
3. Entrenamiento con nuestro dataset y test con nuestro dataset, para estimar el rendimiento del modelo cuando se entrena y evalúa específicamente en el contexto y características de nuestro conjunto de datos.

REFERENCIAS

Temko, Andrey, and Climent Nadeu. 2006. "Classification of Acoustic Events Using SVM-Based Clustering Schemes." *Pattern Recognition* 39: 682–94. <https://doi.org/10.1016/j.patcog.2005.11.005>.