

Temas selectos de ciencia de datos

Tarea 2

Para entregar el 29 de octubre de 2025

Instrucciones.

- Los ejercicios que no son computacionales se entregan en PDF-latex.
- Los ejercicios computacionales se entregan en un solo jupyter notebook por tarea. Incluye cualquier archivo, imagen, datos, etcétera, que se requiera para poder ejecutarse.
- Todos tus entregables debes nombrarlos con el siguiente formato:
`TareaXXX_Nombre_Apellido.ipynb`, `TareaXXX_Nombre_Apellido.pdf`,
`TareaXXX_Nombre_Apellido.csv`, etcétera.
- Si se ejecuta en Colab, asegúrate de dar acceso a los archivos necesarios para ejecutar el código
- Las tareas se entregan en la plataforma Moodle
- Las tareas son individuales, a menos que se especifique lo contrario
- Las indicaciones respecto al uso de IA y demás aspectos éticos a considerar en las tareas y proyectos se dieron al inicio del curso. La calificación se sujetará a tales indicaciones.

1. En clase vimos cómo obtener las frecuencias centrales (en Hz) de diferentes notas o tonalidades. Calcula éstas frecuencias para todas las notas de la escala Do mayor (Figura 1(a)) y Do menor (Figura 1(b)).



Figura 1: (a) Escala C-mayor y (b) escala C-menor. Ambas iniciando en C4.

2. La Figura 2 muestra la forma de onda de un audio de los primeros 8 segundos de la quinta sinfonía de Beethoven (puedes oír el extracto en formato mp3 en el moodle).

 - a) Estima la frecuencia fundamental del sonido registrado en la sección que abarca del segundo 7.3 al 7.8, contando el número de ciclos de oscilación.
 - b) Determina a qué nota musical corresponde como lo vimos en clase, es decir, buscando aquella nota cuya frecuencia fundamental es más cercana a la que estimaste en el inciso anterior. ¿Tiene sentido según la partitura mostrada en la Figura 2?

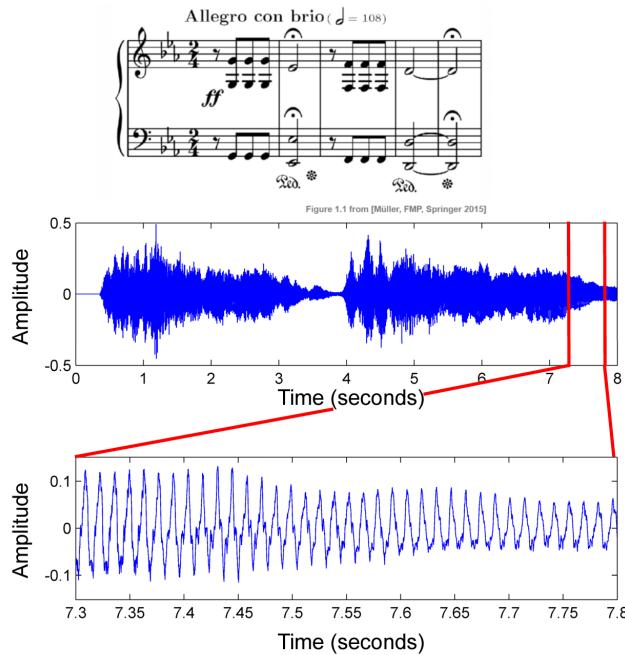


Figura 2: Partitura con los primeros cinco compases de la Sinfonía No. 5 de Beethoven, y su correspondiente señal, como forma de onda, que abarca los primeros 8 segundos.

3. En clase, hablamos sobre el concepto de señal digital y señal analógica, donde ésta última se obtiene mediante dos pasos: muestreo y cuantización de la señal digital.

Este proceso se conoce como *conversión analógica a digital* (ADC: analog-to-digital converter).

Definimos $\mathcal{Q} : \mathbb{R} \rightarrow \Gamma$ como la función de cuantización de una señal continua a un conjunto de valores discretos $\Gamma \subseteq \mathbb{R}$. La función \mathcal{Q} más simple es la cuantización uniforme, donde se asigna a cada valor de amplitud $a \in \mathbb{R}$, un valor cuantizado mediante:

$$\mathcal{Q}(a) = \text{sign}(a)\Delta \left\lfloor \frac{|a|}{\Delta} + \frac{1}{2} \right\rfloor, \quad (1)$$

donde Δ es el tamaño de paso de la cuantización, y $\lfloor \cdot \rfloor$ es la función `floor`. La diferencia entre la señal análoga y la digital obtenida con \mathcal{Q} es el **error de cuantización**.

Muchas veces es más conveniente definir los niveles de cuantización λ en un rango limitado de amplitud $[a_{\min}, a_{\max}]$, en lugar del tamaño de paso Δ . En éste caso, $\Delta = \frac{|a_{\max} - a_{\min}|}{\lambda - 1}$.

- a) Escribe una función que implementa la cuantización uniforme (1)
 - b) Obtén la gráfica de la cuantización (1) para $a = f(x; \theta) \in \mathbb{R}$, con f la función lineal y una sinusoidal con los parámetros que vimos en clase. Grafica también el error absoluto de cuantización correspondiente.
 - c) Con el proceso de cuantización, la señal se *codifica* en alguno de los $\lambda = 2^b$ valores de amplitud igualmente espaciados en el intervalo definido, donde cada intervalo es de tamaño Δ . El parámetro b es el número de bits necesario para codificar la señal, y puede obtenerse mediante $b = \frac{\log \lambda}{\log 2}$. Aunque es posible definir cualquier número de niveles de cuantización para la codificación, es muy común definirlos mediante el número de bits, por ejemplo 8 bits (256 niveles), 16 bits (65536 niveles) o incluso 24 bits (16.8 millones de niveles). Para tener una referencia, el sonido en un CD está codificado generalmente con una cuantización de 16 bits. Realiza la cuantización uniforme de alguna(s) señal(es) de audio que se encuentran en la plataforma Moodle del curso usando 8, 6, 4 y 2 bits. ¿Qué puedes notar en la señal con las diferentes cuantizaciones?
4. Los datos de la carpeta `midi_selected` contiene archivos `midi` de diferentes obras de 8 compositores. Aunque la música se compone de diversos elementos, en éste ejercicio consideraremos sólo 3: notas, acordes y silencios. La función `get_all_notes` del script `fun.py` extrae los acordes, notas y silencios (con su respectiva duración), de los archivos `midi`. El Código MIDI info muestra un ejemplo del uso de la función.
- a) Crea un corpus de las obras musicales de todos los compositores¹. Considera como variable dependiente *y* al compositor, y crea conjuntos de entrenamiento y prueba estratificado por ésta variable de respuesta. ¿Cómo es la distribución de las obras por compositor? Usa algunos gráficos informativos de las características del corpus.

¹Por algunas razones que tienen que ver con errores de transcripción o de formato de los archivos MIDI, algunos archivos resultan vacíos. Puedes eliminar éstos archivos del análisis, o tratar de corregir el error en el script.

- b) Usando una representación TF-IDF² del corpus, implementa al menos 3 clasificadores para estimar y . Utiliza algún método eficiente para el ajuste de los modelos, asegurando una buena generalización. Elabora un resumen breve de tus resultados incluyendo todas las métricas de desempeño, los criterios usados para la representación del corpus, el ajuste de los modelos, tus hallazgos, y demás información que creas pertinente. ¿Qué mejoras sugieres para tener un mejor resultado?

```

1     data_source = '../midi_selected/beethoven/'
2     folder_path = '../midi_selected/beethoven_txt/'
3     metadata = get_all_notes(data_source, folder_path, 'beethoven')
4     metadata.to_csv(os.path.join(data_source, str('beethoven' + \
         '_metadata.csv')), index=False)

```

Código MIDI info: Ejemplo del uso del script para extraer las notas de archivos midi.

5. En la tarea anterior, leíste un ensayo de Daphne Oram. Entonces eres de los pocos afortunados que la conoce, ya que su legado como inventora, compositora y pionera de música electrónica, es (tristemente) desconocido por la mayoría.

Explora un poco más sobre ella y su legado. Puedes ver el contenido de daphneoram.org. Enfócate principalmente en **oramics**, su invento (Figura 3), y describe si lo que has aprendido hasta ahora en el curso, te ayuda a entender y valorar su ingenio. Te recomiendo también ver los videos What is electronic music?, Oramics Machine y Oramics reinvented. Realiza un breve reporte sobre éstos tópicos.



Figura 3: Daphne Oram. Tomada de www.daphneoram.org/oramicsmachine.

²Puedes usar cualquier módulo de Python para vectorizar los archivos, sólo toma en cuenta que el contenido incluye caracteres que algunos tokenizadores los consideran como “separadores de palabras”, por ejemplo, puntos (.), guiones (-) que simbolizan una nota bemol, sostenidos (#), etc. Yo usé el **TfidfVectorizer** de **sklearn**, usando la opción **tokenizer=my_tokenizer**, donde **my_tokenizer** es una función que incluyo en fun.py.