Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 138-150
https://revistas.ulasalle.edu.pe/innosoft

Tipo de artículo: Artículos originales
Temática: Inteligencia Artificial
Recibido: 07/01/2023 | Aceptado: 16/02/2023 | Publicado: 30/03/2023

Identificadores persistentes:
ARK: ark:/42411/s11/a85
PURL: 42411/s11/a85

Creación de un Árbol de Decisión para la Predicción de Tonos a Partir de un Data Set

Analysis of an input dataset to perform a tonal analysis system

Víctor Manuel Vilca Rojas [1][0000-0002-6193-8057]*, **Aldair Bryan Salcedo Chávez** [2][0000-0001-7692-4064], **Jairo Miguel Castillo Rojas** [3][0000-0001-5952-7323] **Valery Byrne Macias** [4][0000-0003-2819-7127]

[1] Universidad Nacional de San Agustín. Arequipa, Perú. vvilcaro@unsa.edu.pe
[2] Universidad Nacional de San Agustín. Arequipa, Perú. asalcedoch@unsa.edu.pe
[3] Universidad Nacional de San Agustín. Arequipa, Perú. jcastillo@unsa.edu.pe
[4] Universidad Nacional de San Agustín. Arequipa, Perú. vbyrne@unsa.edu.pe

* Autor para correspondencia: vvilcaro@unsa.edu.pe

Resumen

El análisis musical es un proceso que se ha llevado a cabo desde hace años donde diferentes expertos han buscado estudiar variadas piezas musicales. Este proceso inicia con el aprendizaje de detección de tonos, notas y acordes, donde los estudiantes tienen que entrenar el oído para poder llevarlo a cabo. Bajo este contexto, en el siguiente trabajo se ha realizado un árbol de decisión en base a un dataset de coros de Bach con el fin de predecir acordes a partir de tonos. Se dividió el dataset en 80% para crear el árbol y 20% para pruebas, después se realizó la transformación de datos para realizar un análisis de los mismos, con esto finalmente se creó un árbol de decisión con una profundidad de 15 y una exactitud del 75.52%, finalmente se realizaron las pruebas y encontramos buenos resultados de la exactitud del árbol.

Palabras clave: Inteligencia artificial, árbol de decisión, análisis musical, detección de tonos, música.

Abstract

Musical analysis is a process that has been carried out for years where different experts have sought to study various musical pieces. This process begins with the learning of tone, note and chord detection, where students have to train their ears to be able to carry it out. In this context, in the following work a decision tree has been made based on a dataset of Bach choirs in order to predict chords from tones. The dataset was divided into 80% to create the tree and 20% for testing, then the data transformation was performed to perform an analysis of the data, with this a decision

tree was finally created with a depth of 15 and an accuracy of 75.52%, the tests were finally carried out and we found good results for the accuracy of the tree.

Keywords: *Artificial intelligence, decision tree, music analysis, pitch detection, music*

Introducción

Las ondas de sonido se propagan a través de varios medios y permiten la comunicación o el entretenimiento para nosotros, los humanos. La música que escuchamos o creamos se puede percibir en aspectos como el ritmo, la melodía, la armonía, el timbre o el estado de ánimo. Todos estos elementos de la música pueden ser de interés para los usuarios de sistemas de recuperación de información musical. Dado que vastos repositorios de música están disponibles para todos en el uso diario (tanto en colecciones privadas como en Internet), es deseable y se vuelve necesario explorar las colecciones de música por contenido. Por lo tanto, la recuperación de información musical puede ser potencialmente de interés para todos los usuarios de computadoras e Internet [1].

Dado un flujo musical, la tarea del análisis de la armonía musical consiste en asociar una etiqueta a cada punto de tiempo. Tales etiquetas revelan la armonía subyacente al indicar una nota fundamental (raíz) y un modo, usando nombres de acordes como 'Do menor'. La tarea de análisis musical puede representarse naturalmente como un problema de aprendizaje secuencial supervisado. De hecho, al considerar sólo las clases de tonos actualmente resonantes, difícilmente se producirían análisis razonables. Las evidencias experimentales sobre la cognición humana revelan que para eliminar la ambigüedad de los casos poco claros, los compositores y los oyentes también se refieren a las transiciones de acordes: en estos casos, el contexto juega un papel fundamental y las claves contextuales pueden ser útiles para el análisis [6].

El dominio musical siempre ha ejercido una fuerte fascinación sobre investigadores de diversos campos. En los últimos años se ha invertido un gran esfuerzo de investigación para analizar la música, bajo una presión académica e industrial. Las técnicas de búsqueda y análisis de música inteligente son cruciales para diseñar sistemas para varios propósitos, como la identificación de música, para decidir sobre la similitud de la música, para la clasificación de música basada en algún conjunto de descriptores, para la generación algorítmica de listas de reproducción y para el resumen de música. De hecho, los avances tecnológicos recientes mejoraron significativamente la forma en que los entornos automáticos

componen música, la interpretan expresivamente, acompañan a músicos humanos y la forma en que se vende y compra música a través de tiendas web [5].

El análisis musical es un paso necesario para componer, interpretar y en última instancia, comprender la música, tanto para los seres humanos como para los entornos artificiales. Dentro del área más amplia del análisis musical, destacamos la tarea del reconocimiento de acordes. Este es un problema desafiante para los estudiantes de música, que dedican una cantidad considerable de tiempo a aprender la armonía tonal, así como para los sistemas automáticos. Es un problema interesante y un paso necesario para realizar un análisis estructural de alto nivel que considere los principales elementos estructurales de la música en sus interconexiones mutuas. En la música tonal occidental, en cada momento del flujo musical (o vertical) se puede determinar qué acorde está sonando: el reconocimiento de acordes normalmente consiste en indicar la nota fundamental (o raíz) y el modo del acorde (Figura 1) [5].

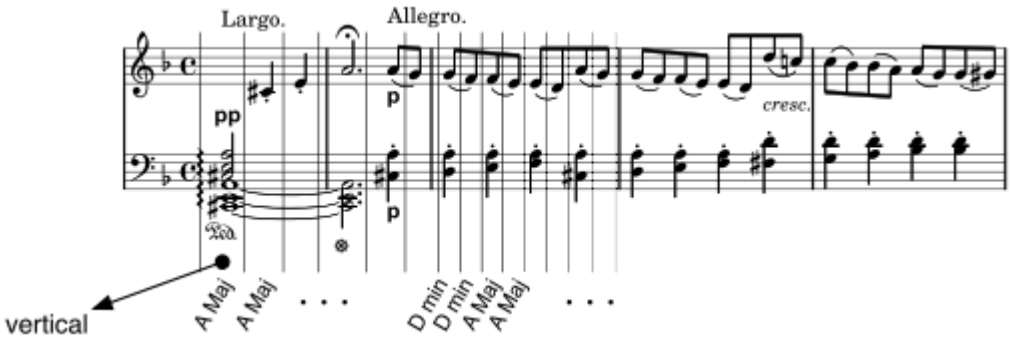


Figura 1 :El problema de reconocimiento de acordes consiste en indicar para cada vertical qué acorde está sonando en ese momento. Extracto de la Sonata para piano Opus 31 n.2 de Beethoven, 1er movimiento.

En síntesis, como dice Bent [7], el análisis musical consiste en exponer y describir sintéticamente la estructura musical y la manera en que se relacionan esos elementos más simples en la estructura general. El análisis puede ir desde una parte de una pieza musical hasta una colección de obras en un periodo de tiempo exacto. Pero el proceso seguirá siendo el mismo utilizando tradicionalmente el oído, papel y un lápiz [8]. Sin embargo, si nos proyectamos a un ámbito más actual, donde frente a nosotros no tenemos una partitura sino una colección de CD´s o hasta un celular con una lista de

reproducción de Spotify, las técnicas anteriormente utilizadas para analizar este contenido resultan difíciles de usar, por no decir inservibles.

En estas situaciones del mundo moderno es que surgen diferentes métodos acompañados de algoritmos, los cuales participan en nuevos sistemas de análisis [9].

En este párrafo se expondrá los métodos a utilizar en este artículo: como primer punto se empleará la técnica de árbol de decisión, ya que es una técnica de forma gráfica y analítica de representar todos los eventos o sucesos que pueden surgir a partir de una decisión asumida en cierto momento. Esto nos permitirá tomar la decisión más acertada, ante una variedad de posibles decisiones [10]. Cómo segunda técnica a emplear tenemos clustering, esta técnica es utilizada como un proceso para encontrar una estructura significativa, procesos subyacentes explicativos, características generativas y agrupaciones inherentes a un conjunto de ejemplos; lo que la hace idónea para la presente investigación [11]. Entre otras técnicas de minería de datos.

Entonces, ya habiendo comprendido las bases teóricas de esta investigación, explicaremos la finalidad de este artículo, el cual es: “Desarrollar el análisis del dataset de entrada para realizar un sistema de análisis tonal” el cual nos dará como resultado, un acorde después de recibir los siguientes datos de ingreso: archivos de Bach Central [WebLink]; donde se reconocerá las notas musicales con un SI/NO dependiendo en qué lugar se presente el tono.

Herramientas

Árbol de decisiones: Se utilizó árboles de decisiones ya que es una manera de representación de todos los eventos que pueden surgir por una decisión asumida en cierto momento la cual puede ser gráfica y analítica que nos permite organizar el trabajo de cálculos correspondientes para así un despliegue visual del problema. Además de ello nos ayuda a tomar la decisión más idónea, desde el punto de vista probabilístico de un sin fin de posibles decisiones [17].

Correlación de Pearson: La correlación de Pearson mide la existencia (dada por un valor p) y la fuerza (dada por el coeficiente r entre -1 y +1) de una relación lineal entre dos variables continuas. Solo debe usarse cuando

se satisfacen sus supuestos subyacentes. Si el resultado es significativo, concluimos que existe una correlación. Para mejor entendimiento un valor absoluto de r de 0,1 se clasifica como pequeño, un valor absoluto de 0,3 se clasifica como medio y de 0,5 se clasifica como grande [18].

Métodos y Metodología computacional

Google Colab: Es un entorno colaborativo de Google que permite trabajar con Notebooks sin alguna configuración requerida que se ejecuta en la nube y te proporciona acceso gratuito a GPU lo cual te permite escribir y ejecutar código de Python en tu navegador y que a su vez permite almacenar dichos cuadernos y trabajar con datos que tengas almacenados en el Drive y compartirlos con tu equipo de trabajo [12].

Python: Es un lenguaje de programación de alto nivel, además es un lenguaje interpretado, por lo cual no requiere ser compilado. Por este motivo el programador puede utilizar el lenguaje de forma directa en el programa o aplicación que realice. Una de las características más resaltantes es que cuenta con una amplia bibliotecas de librerías que permiten obtener diversos recursos de código abierto aplicables para la inteligencia artificial [13].

NumPy: Es una biblioteca idónea para la manipulación de muchos datos. El problema yace en que Python usa listas y no arreglos por ello NumPy nos provee estructuras muy eficientes para manipular muchos datos [14].

Pandas: Es un buen toolkit para hacer análisis de datos. Tiene herramientas para tener tablas y otras estructuras de datos. Además de ello nos permite cargar con gran facilidad archivos csv [15].

Seaborn: Es una librería que usa Matplotlib por debajo para trazar gráficos. Lo cual será usado para visualizar distribuciones aleatorias [16].

Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 138-150
https://revistas.ulasalle.edu.pe/innosoft

Trabajos relacionados

Si vemos en un entorno específico, dentro de la enseñanza de la música, las TICs (Tecnologías de la Información y la Comunicación) aportaron al proceso de enseñanza-aprendizaje. Por un lado, ayudaron a los maestros que tenían alrededor 15 alumnos a personalizar más sus sesiones y por otro lado, el estudiante podía solventar sus dudas en cualquier momento al interactuar con este sistema. Estas TICs como herramientas comenzaron a complementar las sesiones en diferentes conservatorios, que cabe recalcar que solo se llevaban una vez a la semana, y de tal modo el descenso del nivel de conocimiento de los estudiantes de años actuales, a comparación de años anteriores, comenzó a menguar [9].

Siguiendo esta investigación, Illescas logró desarrollar un software que realizaba análisis musical, el cual orientado a la pedagogía lo probó en el Conservatorio Superior de Música de Murcia. Para realizar una buena interpretación de una pieza, se necesita realizar un buen análisis de la partitura, por lo que este software demostró su ayuda tanto como para alumnos y como para maestros [9].

Resultados y discusión

Inicialmente se importan las librerías necesarias para ejecutar nuestro árbol de decisiones.

```
# Imports necesarios
import numpy as np
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
%matplotlib inline
plt.rcParams['figure.figsize'] = (16, 9)
plt.style.use('ggplot')
from sklearn import tree
from sklearn.metrics import accuracy_score
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from IPython.display import Image as PImage
from subprocess import check_call
from PIL import Image, ImageDraw, ImageFont
from sklearn.model_selection import train_test_split
```

Cargamos los valores de entrada.

Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 138-150
https://revistas.ulasalle.edu.pe/innosoft

```
[ ] bach_choral = pd.read_csv(r"bach_choral_set_dataset.csv")

[ ] from google.colab import drive
    drive.mount('/content/drive')

    Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

[ ] bach_choral.shape

    (4532, 17)

[ ] bach_choral.head()
```

	choral_ID	event_number	pitch_1	pitch_2	pitch_3	pitch_4	pitch_5	pitch_6	pitch_7	pitch_8	pitch_9	pitch_10	pitch_11	pitch_12	bass	meter	chord_label
0	000106b_	1	YES	NO	NO	NO	NO	YES	NO	NO	NO	YES	NO	NO	F	3	F_M
1	000106b_	2	YES	NO	NO	NO	YES	NO	NO	YES	NO	NO	NO	NO	E	5	C_M
2	000106b_	3	YES	NO	NO	NO	YES	NO	NO	YES	NO	NO	NO	NO	E	2	C_M
3	000106b_	4	YES	NO	NO	NO	NO	YES	NO	NO	NO	YES	NO	NO	F	3	F_M
4	000106b_	5	YES	NO	NO	NO	NO	YES	NO	NO	NO	YES	NO	NO	F	2	F_M

Para ello se usó Data Transformation para cambiar los valores “NO” y “YES”

Los valores respectivos son 0 = no, 1 = sí

```
bach_choral.groupby('chord_label').size()

chord_label
A#d       3
A#d7      2
A_M     235
A_M4     11
A_M6      2
        ...
G_M6      2
G_M7     48
G_m     153
G_m6      3
G_m7     18
Length: 95, dtype: int64

[ ] categorical_cols = ['bass','chord_label']
    for column in categorical_cols:
        bach_choral[column] = pd.factorize(bach_choral[column])[0]

    for i in range(1,13):
        pitchChange = "pitch_" + str(i)
        bach_choral[pitchChange] = bach_choral[pitchChange].replace('YES', 1)
        bach_choral[pitchChange] = bach_choral[pitchChange].replace('NO', 0)

    drop_elements = ['choral_ID','event_number']
    bach_choral_encoded = bach_choral.drop(drop_elements, axis = 1)

    bach_choral_encoded
```

Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 138-150
https://revistas.ulasalle.edu.pe/innosoft

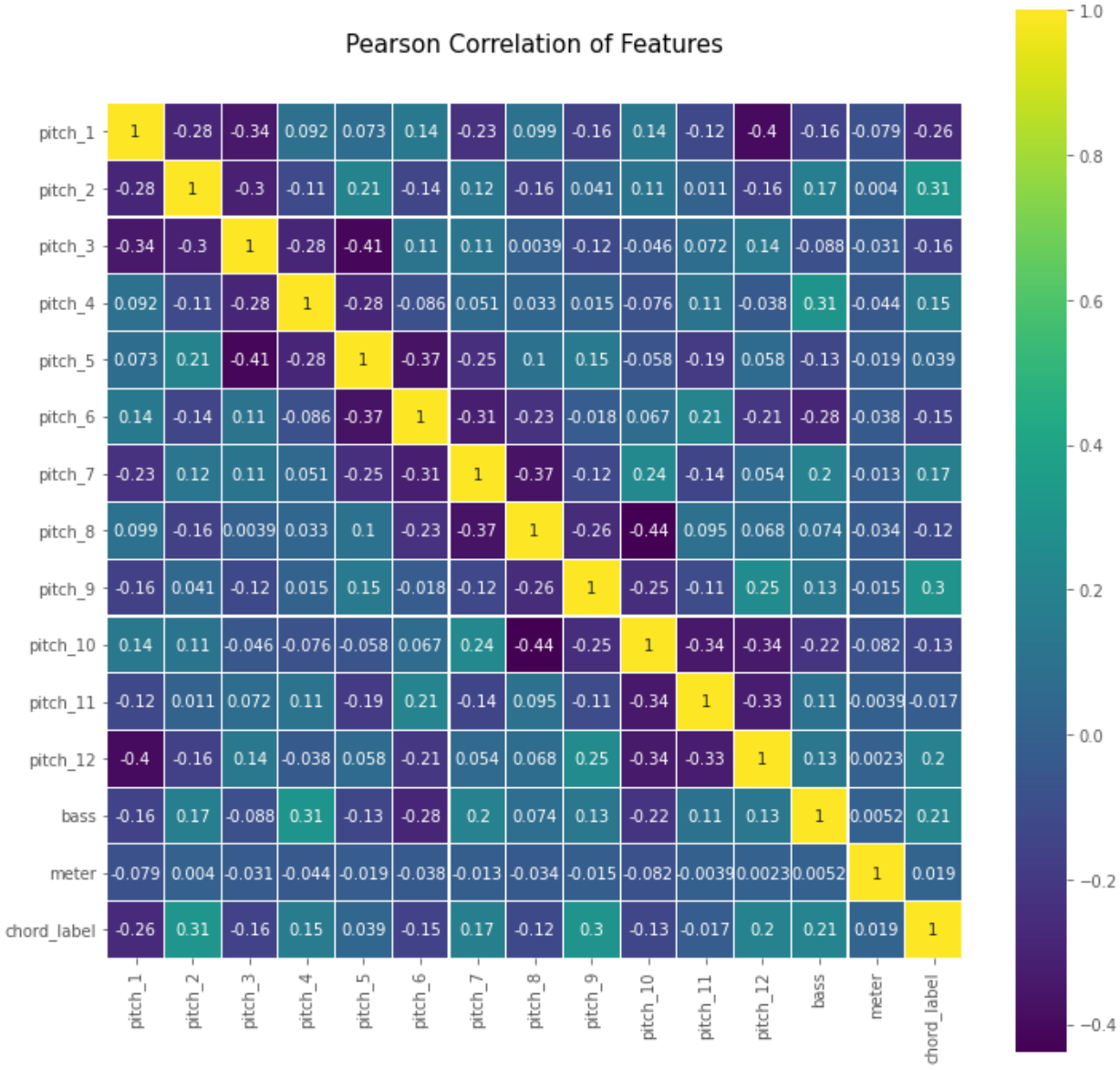
	pitch_1	pitch_2	pitch_3	pitch_4	pitch_5	pitch_6	pitch_7	pitch_8	pitch_9	pitch_10	pitch_11	pitch_12	bass	meter	chord_label
0	1	0	0	0	0	1	0	0	0	1	0	0	0	3	0
1	1	0	0	0	1	0	0	1	0	0	0	0	1	5	1
2	1	0	0	0	1	0	0	1	0	0	0	0	1	2	1
3	1	0	0	0	0	1	0	0	0	1	0	0	0	3	0
4	1	0	0	0	0	1	0	0	0	1	0	0	0	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4527	0	1	0	0	1	0	0	1	0	1	0	0	9	3	11
4528	0	0	1	0	0	0	1	0	0	1	0	0	2	4	11
4529	0	0	1	0	0	0	1	0	0	1	0	0	2	3	11
4530	0	0	1	0	1	0	0	1	0	1	0	0	1	2	11
4531	0	0	1	0	0	0	1	0	0	1	0	0	8	5	11

4532 rows × 15 columns

Luego de ello se realizó un análisis de nuestros datos de entrada categóricos el cual mostrará una correlación de Pearson en su mayoría pequeña(o débil) entre los datos ya que la mayoría ésta inferior al 0.3.

```
[ ]  colormap = plt.cm.viridis
     plt.figure(figsize=(12,12))
     plt.title('Pearson Correlation of Features', y=1.05, size=15)
     sb.heatmap(bach_choral_encoded.astype(float).corr(),linewidths=0.1,vmax=1.0, square=True, cmap=colormap, linecolor='white', annot=True)
```

Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 138-150
https://revistas.ulasalle.edu.pe/innosoft



Para continuar se realiza la creación del Árbol Decisión

```
# Crear arrays de entrenamiento y las etiquetas que indican si llegó a top o no
y_train = bach_choral_encoded['chord_label']
x_train = bach_choral_encoded.drop(['chord_label'], axis=1).values
```

Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 138-150
https://revistas.ulasalle.edu.pe/innosoft

```
# Crear Arbol de decision con profundidad = 4
decision_tree = tree.DecisionTreeClassifier(criterion='entropy',
                                            min_samples_split=20,
                                            min_samples_leaf=5,
                                            max_depth = 15,
                                            class_weight={1:3.5})
decision_tree.fit(x_train, y_train)
```

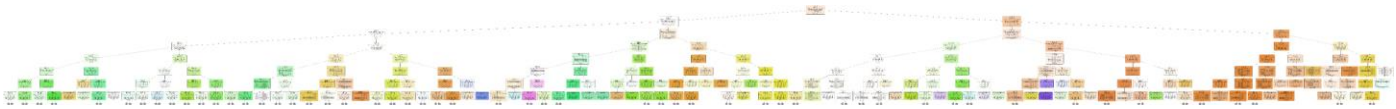
```
# exportar el modelo a archivo .dot
with open(r"tree1.dot", 'w') as f:
    f = tree.export_graphviz(decision_tree,
                             out_file=f,
                             max_depth = 7,
                             impurity = True,
                             feature_names = list(bach_choral_encoded.drop(['chord_label'], axis=1)),
                             class_names = ['F_M',
'C_M',
'D_m',
'BbM',
'C_M7',
'D_m7',
'G_M',
'A_m',
'C_M4',
'G_m',
'G_M7',
'D_M',
'F#d',
'AbM',
'C#d7',
'D_M7',
'A_M',
'EbM',
'F_M7',
```

```
'Bbd',
'Dbm7',
'Abm',
'DbM7',
'Dbm',
'F#m6',
'G#m',
'B_d',
'C_M6',
'D#m',
'D#M',
'BbM7',
'F_d7',
'C#d6',
'G_d',
'G#M',
'C#M4',
'D#d6',
'D#d7'],
                                rounded = True,
                                filled= True )

# Convertir el archivo .dot a png para poder visualizarlo
check_call(['dot','-Tpng',r'tree1.dot','-o',r'tree1.png'])
PImage("tree1.png")
```

Revista Innovación y Software
Vol. 4, No. 1, Mes Marzo-Agosto, 2023
ISSN: 2708-0935
Pág. 138-150
https://revistas.ulasalle.edu.pe/innosoft

Obtenemos el siguiente árbol:



Se realiza el mapeo de atributos y las predicciones del árbol de decisión.

```
Prediccion: [11]
Probabilidad de Acierto: 81.82%
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:444: UserWarning: X has feature names, but DecisionTreeClassifier was fitted without feature names
  f"X has feature names, but {self.__class__.__name__} was fitted without"
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:444: UserWarning: X has feature names, but DecisionTreeClassifier was fitted without feature names
  f"X has feature names, but {self.__class__.__name__} was fitted without"
```

```
#predecir test data

x_test = pd.read_csv(r"bach_choral_set_test.csv")
categorical_cols = ['bass','chord_label']
for column in categorical_cols:
    x_test[column] = pd.factorize(x_test[column])[0]

for i in range(1,13):
    pitchChange = "pitch_" + str(i)
    x_test[pitchChange] = x_test[pitchChange].replace('YES', 1)
    x_test[pitchChange] = x_test[pitchChange].replace('NO', 0)

drop_elements = ['choral_ID','event_number']
x_test_encoded = x_test.drop(drop_elements, axis = 1)

x_test_encoded
```

	pitch_1	pitch_2	pitch_3	pitch_4	pitch_5	pitch_6	pitch_7	pitch_8	pitch_9	pitch_10	pitch_11	pitch_12	bass	meter	chord_label
0	0	0	1	0	0	0	1	1	0	1	0	0	0	2	0
1	0	0	1	0	0	0	1	0	0	0	0	1	1	3	1
2	0	0	0	0	1	0	0	1	0	0	0	1	2	4	2
3	0	0	0	0	1	0	0	1	0	0	0	1	2	3	2
4	0	0	1	0	1	0	0	1	0	0	0	1	3	2	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1128	0	0	1	0	0	0	0	1	0	0	1	0	6	4	44
1129	0	0	1	0	0	0	0	1	0	1	0	0	6	3	44
1130	1	0	0	0	1	0	0	1	0	0	0	0	8	5	7
1131	1	0	0	0	1	0	0	1	0	0	1	0	8	3	7
1132	0	0	0	0	0	1	0	0	0	1	0	0	7	4	40

1133 rows × 15 columns

Imprimimos las predicciones

```
y_pred = decision_tree.predict(x_test_encoded.drop(['chord_label'], axis = 1))
print("Prediccion: " + str(y_pred))
y_proba = decision_tree.predict_proba(x_test_encoded.drop(['chord_label'], axis = 1))
print("Probabilidad de Acierto: " + str(round(y_proba[0][y_pred][0]* 100, 2))+"%")
```

Conclusiones

Se puede concluir que nuestro modelo de clasificación genera modelos acertados con un 75.52% de exactitud. Esto nos indica que nuestro árbol de decisión es capaz de predecir correctamente la nota musical en el 75.52% de los casos, esto a partir de los datos de entrada que consideramos en este trabajo así como las características que indicamos para la creación del árbol. También encontramos que nuestro dataset, al tratarse de notas musicales de distintas canciones, y ademas, teniendo en cuenta que los eventos del dataset están numerados consecutivamente, indicándonos el orden en que estos hacen su aparición en la canción, es que podemos entender porque la correlación entre las variables de entrada es tan baja, una mejor opción sería realizar otros análisis de correlación. Además, en la búsqueda de un mayor grado de exactitud del árbol de decisión, encontramos que la profundidad más adecuada para este era de 15, un mínimo de hojas para dividir el nodo de 20 y un mínimo de eventos para considerar hoja de 5, además, al aumentar el peso de la clase pitch_1 del dataset de 1 a 3.5 con una diferencia de 2.5 respecto a las demás clases, encontramos que el árbol de decisión incrementó su exactitud en un porcentaje mayor al 30%.

Como trabajo futuro. Teniendo ya la base del conocimiento de nuestro proyecto se puede realizar un identificador de acordes para cualquier artista, el cual con un dataset de sus pistas podemos deducir cuales son los acordes más tocados en sus pistas.

Referencias

[1] Raś, Z. W., & Wieczorkowska, A. A. (Eds.). (2010). Advances in Music Information Retrieval. Studies in Computational Intelligence. doi:10.1007/978-3-642-11674-2

[2] R. Esposito and D. P. Radicioni, “CarpeDiem: Optimizing the viterbi algorithm and applications to supervised sequential learning,” *J. Mach. Learn. Res.*, vol. 10, pp. 1851–1880, 2009.

[3] M. Rohrmeier and I. Cross, “Statistical Properties of Tonal Harmony in Bach ’ s Chorales Statistical Properties of Tonal Harmony in Bach ’ s Chorales,” no. January 2008, 2014.

[4] D. P. Radicioni and R. Esposito, “Learning tonal harmony from Bach chorales,” *Procs. 7th Int. Conf. Cogn. Model.*, no. August, 2006.

[5] D. P. Radicioni and R. Esposito, “BREVE: An HMPerceptron-based chord recognition system,” *Stud. Comput. Intell.*, vol. 274, no. December, pp. 143–164, 2010, doi: 10.1007/978-3-642-11674-2_7.

[6] D. P. Radicioni and R. Esposito, "UCI Machine Learning Repository: Bach Choral Harmony Data Set", Archive.ics.uci.edu, 2014. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Bach+Choral+Harmony#. [Accessed: 20- Jun- 2022].

[7] I. Bent. "Music analysis in the nineteenth century" 1994.

[8] Federico Sammartino. "Ceros y unos en la musicología. Software y análisis musical". http://resonancias.uc.cl/images/PDFs_n_37/Sammartino.pdf.

[9] P. R. Illescas Casanova. "Análisis tonal asistido por ordenador". Dialnet. https://dialnet.unirioja.es/servlet/tesis?codigo=60733 .

[10] V. Berlanga-Silvente, M. J. Rubio-Hurtado, and R. Vilà-Baños, “Cómo aplicar árboles de decisión en SPSS,” REIRE. Rev. d’Innovació i Recer. en Educ., vol. 6, no. 1, pp. 65–79, 2013, doi: 10.1344/reire2013.6.1615.

[11] D. Tamm, “Road Map,” Dtsch. Arztebl. Int., vol. 115, no. 35–36, p. A1554, 2018, doi: 10.4324/9781003191056-1.

[12] “Te damos la bienvenida a Colab.” https://colab.research.google.com/?utm_source=scs-index.

[13] “Python 3.10.6 documentation,” [Online]. Available: https://docs.python.org/3/.

[14] “NumPy.” https://numpy.org/.

[15] “Pandas.” https://pandas.pydata.org/.

[16] “Seaborn.” https://www.w3schools.com/python/numpy/numpy_random_seaborn.asp.

[17] M. C. Ruiz Abellón, “Introducción a los árboles de decisión,” pp. 1–7, 2014, [Online]. Available: http://www.dmae.upct.es/~mcruiz/Telem06/Teoria/arbol_decision.pdf.

[18] P. Samuels, “וסריפ מאתמ Pearson Correlation ? מאתמ והמ,” no. April 2014, pp. 1–5, 2015, [Online]. Available: https://www.researchgate.net/publication/274635640.

Roles de Autoría

Víctor Manuel Vilca Rojas: Conceptualización, Curación de datos, Análisis formal, Investigación, Metodología, Software, Validación, Redacción - borrador original. **Aldair Bryan Salcedo Chávez:** Conceptualización, Curación de datos, Análisis formal, Investigación, Metodología, Software, Validación, Redacción - borrador original. **Jairo Miguel Castillo Rojas:** Conceptualización, Curación de datos, Análisis formal, Investigación, Metodología, Software, Validación, Redacción - borrador original. **Valery Byrne Macias:** Conceptualización, Curación de datos, Análisis formal, Investigación, Metodología, Software, Validación, Redacción - borrador original.