Taller Machine learning

Presentado por:

Juan David Mosquera

Arley Paniagua

Docente: Juan Camilo

Programa: Ingeniería Informática

Universidad Autónoma Latinoamericana

23 de octubre de 2023

Introducción:

El trabajo desarrollado consistía en un script en Python que realiza un análisis de datos y utiliza un clasificador de árbol de decisión para predecir la música de acuerdo con algunas características de las canciones.

Desarrollo:

Para implementar el trabajo lo primero que se hace es importar las librerías necesarias, se importan varias bibliotecas de Python necesarias para el análisis de datos y la construcción del modelo de clasificación. Estas bibliotecas incluyen pandas para la manipulación de datos, “DecisionTreeClassifier” de “sklearn.tree” para el clasificador de árbol de decisión, y otras bibliotecas relacionadas con la visualización de árboles.

Posteriormente se cargan los datos del data set sobre los cuales se va a trabajar.

	Unnamed: 0	instance_id	artist_name	track_name	popularity	acousticness	danceability	duration_ms	energy
0	1	46652.0	Thievery Corporation	The Shining Path	31.0	0.01270	0.622	218293.0	0.890
1	2	30097.0	Dillon Francis	Hurricane	28.0	0.00306	0.620	215613.0	0.755
2	3	62177.0	Dubloadz	Nitro	34.0	0.02540	0.774	166875.0	0.700
3	4	24907.0	What So Not	Divide & Conquer	32.0	0.00465	0.638	222369.0	0.587

Se define una lista de características, los datos se dividen en conjuntos de entrenamiento y prueba. Donde se tiene un 70% para entrenar y un 30% para probar el modelo de acuerdo a la siguiente línea.

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=220)
```

Se crea un clasificador de árbol de decisión y se entrena utilizando los datos de entrenamiento. Se utiliza el modelo entrenado para hacer predicciones en el conjunto de prueba. En este paso además se define el máximo de niveles que podrá tener el árbol en este caso son 10.

```
classifier = DecisionTreeClassifier(criterion="entropy", max_depth=10)
classifier = classifier.fit(X_train, Y_train)
Y_pred = classifier.predict(X_test)
Y_pred
```

```
array([58., 37., 61., ..., 89., 54., 58.])
```

Se calcula la precisión del modelo comparando las predicciones con las etiquetas reales. El resultado se imprime en pantalla.

```
print("Accuracy:", metrics.accuracy_score(Y_test, Y_pred))
```

```
Accuracy: 0.9971236028928336
```

En este caso la precisión es de 0,99% lo cual significa que nuestro modelo tiene una alta tasa de aciertos, en realidad es casi perfecto.

Lo último que se hace es crear el árbol para graficar este árbol de decisión es un modelo de clasificación breve que funciona de la siguiente manera interpretación de cómo funciona el árbol:

- El árbol comienza dividiendo las canciones en dos grupos basándose en la característica "popularity". Si la po pularidad es menor o igual a 44.50, el árbol sigue una serie de divisiones adicionales para predecir un valor es pecífico de popularidad. Si la popularidad es mayor que 44.50, el árbol también sigue una serie de divisiones p ara predecir un valor específico de popularidad.

- Cada nodo del árbol representa una característica (en este caso, la "popularity" en diferentes intervalos) y un valor de corte que se utiliza para tomar decisiones.

- El valor "class" en los nodos hoja representa la predicción de la popularidad de las canciones en ese grupo. P or ejemplo, si llegas al nodo "class: 10.0", significa que, según las características de la canción evaluada, se pre dice que su popularidad será 10.0.

- El árbol se ramifica en función de las características de las canciones y de los valores de popularidad en el co njunto de datos de entrenamiento. Cada nodo padre se divide en nodos hijos en función de un criterio de decis ión, como la entropía. Que el caso presente se estableció en 10.

Una vez este árbol está trabajando correctamente el funcionamiento es simple si tienes una nueva canción y d eseas predecir su popularidad, puedes recorrer el árbol desde la raíz hasta una hoja para obtener la predicció n. Por ejemplo, si la popularidad de una canción es menor o igual a 26.50, el árbol predice que su popularidad será 13.0.
A continuación se muestra el árbol generado en formato de texto:

Árbol de decisión en formato de texto:

```
|--- popularity <= 44.50
|   |--- popularity <= 33.50
|   |   |--- popularity <= 26.50
|   |   |   |--- popularity <= 19.50
|   |   |   |   |--- popularity <= 13.50
|   |   |   |   |   |--- popularity <= 3.50
|   |   |   |   |   |   |--- popularity <= 0.50
|   |   |   |   |   |   |   |--- class: 0.0
|   |   |   |   |   |   |--- popularity >  0.50
|   |   |   |   |   |   |   |--- popularity <= 1.50
|   |   |   |   |   |   |   |   |--- class: 1.0
|   |   |   |   |   |   |--- popularity >  1.50
|   |   |   |   |   |   |   |--- popularity <= 2.50
|   |   |   |   |   |   |   |   |--- class: 2.0
|   |   |   |   |   |   |--- popularity >  2.50
|   |   |   |   |   |   |   |--- class: 3.0
|   |   |   |   |   |--- popularity >  3.50
|   |   |   |   |   |   |--- popularity <= 11.50
|   |   |   |   |   |   |   |--- popularity <= 9.50
|   |   |   |   |   |   |   |   |--- popularity <= 7.50
|   |   |   |   |   |   |   |   |   |--- popularity <= 4.50
|   |   |   |   |   |   |   |   |   |   |--- class: 4.0
|   |   |   |   |   |   |   |   |   |--- popularity >  4.50
|   |   |   |   |   |   |   |   |   |   |--- class: 5.0
|   |   |   |   |   |   |   |   |--- popularity >  7.50
|   |   |   |   |   |   |   |   |   |--- popularity <= 8.50
|   |   |   |   |   |   |   |   |   |   |--- class: 8.0
|   |   |   |   |   |   |   |   |   |--- popularity >  8.50
|   |   |   |   |   |   |   |   |   |   |--- class: 9.0
|   |   |   |   |   |   |   |--- popularity >  9.50
|   |   |   |   |   |   |   |   |--- popularity <= 10.50
|   |   |   |   |   |   |   |   |   |--- class: 10.0
|   |   |   |   |   |   |   |   |--- popularity >  10.50
|   |   |   |   |   |   |   |   |   |--- class: 11.0
|   |   |   |   |   |   |--- popularity >  11.50
|   |   |   |   |   |   |   |--- popularity <= 12.50
|   |   |   |   |   |   |   |   |--- class: 12.0
|   |   |   |   |   |   |--- popularity >  12.50
|   |   |   |   |   |   |   |--- class: 13.0
|   |   |   |   |--- popularity >  13.50
|   |   |   |   |   |--- popularity <= 16.50
|   |   |   |   |   |   |--- popularity <= 15.50
|   |   |   |   |   |   |   |--- popularity <= 14.50
|   |   |   |   |   |   |   |   |--- class: 14.0
|   |   |   |   |   |   |   |--- popularity >  14.50
|   |   |   |   |   |   |   |   |--- class: 15.0
|   |   |   |   |   |   |--- popularity >  15.50
```

```
|  |  |  |  |  |  |  |--- class: 16.0
|  |  |  |  |  |--- popularity >  16.50
|  |  |  |  |  |  |--- popularity <= 18.50
|  |  |  |  |  |  |  |--- popularity <= 17.50
|  |  |  |  |  |  |  |  |--- class: 17.0
|  |  |  |  |  |  |  |--- popularity >  17.50
|  |  |  |  |  |  |  |  |--- class: 18.0
|  |  |  |  |  |  |--- popularity >  18.50
|  |  |  |  |  |  |  |--- class: 19.0
|  |  |  |--- popularity >  19.50
|  |  |  |  |--- popularity <= 23.50
|  |  |  |  |  |--- popularity <= 21.50
|  |  |  |  |  |  |--- popularity <= 20.50
|  |  |  |  |  |  |  |--- class: 20.0
|  |  |  |  |  |  |--- popularity >  20.50
|  |  |  |  |  |  |  |--- class: 21.0
|  |  |  |  |  |--- popularity >  21.50
|  |  |  |  |  |  |--- popularity <= 22.50
|  |  |  |  |  |  |  |--- class: 22.0
|  |  |  |  |  |  |--- popularity >  22.50
|  |  |  |  |  |  |  |--- class: 23.0
|  |  |  |  |--- popularity >  23.50
|  |  |  |  |  |--- popularity <= 25.50
|  |  |  |  |  |  |--- popularity <= 24.50
|  |  |  |  |  |  |  |--- class: 24.0
|  |  |  |  |  |  |--- popularity >  24.50
|  |  |  |  |  |  |  |--- class: 25.0
|  |  |  |  |  |--- popularity >  25.50
|  |  |  |  |  |  |--- class: 26.0
|  |  |--- popularity >  26.50
|  |  |  |--- popularity <= 30.50
|  |  |  |  |--- popularity <= 28.50
|  |  |  |  |  |--- popularity <= 27.50
|  |  |  |  |  |  |--- class: 27.0
|  |  |  |  |  |--- popularity >  27.50
|  |  |  |  |  |  |--- class: 28.0
|  |  |  |  |--- popularity >  28.50
|  |  |  |  |  |--- popularity <= 29.50
|  |  |  |  |  |  |--- class: 29.0
|  |  |  |  |  |--- popularity >  29.50
|  |  |  |  |  |  |--- class: 30.0
|  |  |  |--- popularity >  30.50
|  |  |  |  |--- popularity <= 32.50
|  |  |  |  |  |--- popularity <= 31.50
|  |  |  |  |  |  |--- class: 31.0
|  |  |  |  |  |--- popularity >  31.50
|  |  |  |  |  |  |--- class: 32.0
|  |  |  |  |--- popularity >  32.50
|  |  |  |  |  |--- class: 33.0
|  |--- popularity >  33.50
|  |  |--- popularity <= 38.50
|  |  |  |--- popularity <= 36.50
|  |  |  |  |--- popularity <= 35.50
|  |  |  |  |  |--- popularity <= 34.50
|  |  |  |  |  |  |--- class: 34.0
```

```
|  |  |  |  |  |--- popularity >  34.50
|  |  |  |  |  |  |--- class: 35.0
|  |  |  |  |--- popularity >  35.50
|  |  |  |  |  |--- class: 36.0
|  |  |  |--- popularity >  36.50
|  |  |  |  |--- popularity <= 37.50
|  |  |  |  |  |--- class: 37.0
|  |  |  |  |--- popularity >  37.50
|  |  |  |  |  |--- class: 38.0
|  |  |--- popularity >  38.50
|  |  |  |--- popularity <= 41.50
|  |  |  |  |--- popularity <= 39.50
|  |  |  |  |  |--- class: 39.0
|  |  |  |  |--- popularity >  39.50
|  |  |  |  |  |--- popularity <= 40.50
|  |  |  |  |  |  |--- class: 40.0
|  |  |  |  |  |--- popularity >  40.50
|  |  |  |  |  |  |--- class: 41.0
|  |  |  |--- popularity >  41.50
|  |  |  |  |--- popularity <= 42.50
|  |  |  |  |  |--- class: 42.0
|  |  |  |  |--- popularity >  42.50
|  |  |  |  |  |--- popularity <= 43.50
|  |  |  |  |  |  |--- class: 43.0
|  |  |  |  |  |--- popularity >  43.50
|  |  |  |  |  |  |--- class: 44.0
|--- popularity >  44.50
|  |--- popularity <= 55.50
|  |  |--- popularity <= 50.50
|  |  |  |--- popularity <= 47.50
|  |  |  |  |--- popularity <= 46.50
|  |  |  |  |  |--- popularity <= 45.50
|  |  |  |  |  |  |--- class: 45.0
|  |  |  |  |  |--- popularity >  45.50
|  |  |  |  |  |  |--- class: 46.0
|  |  |  |  |--- popularity >  46.50
|  |  |  |  |  |--- class: 47.0
|  |  |  |--- popularity >  47.50
|  |  |  |  |--- popularity <= 49.50
|  |  |  |  |  |--- popularity <= 48.50
|  |  |  |  |  |  |--- class: 48.0
|  |  |  |  |  |--- popularity >  48.50
|  |  |  |  |  |  |--- class: 49.0
|  |  |  |  |--- popularity >  49.50
|  |  |  |  |  |--- class: 50.0
|  |  |--- popularity >  50.50
|  |  |  |--- popularity <= 52.50
|  |  |  |  |--- popularity <= 51.50
|  |  |  |  |  |--- class: 51.0
|  |  |  |  |--- popularity >  51.50
|  |  |  |  |  |--- class: 52.0
|  |  |  |--- popularity >  52.50
|  |  |  |  |--- popularity <= 53.50
|  |  |  |  |  |--- class: 53.0
|  |  |  |  |--- popularity >  53.50
```

```
|  |  |  |  |  |--- popularity <= 54.50
|  |  |  |  |  |  |--- class: 54.0
|  |  |  |  |  |--- popularity >  54.50
|  |  |  |  |  |  |--- class: 55.0
|  |--- popularity >  55.50
|  |  |--- popularity <= 61.50
|  |  |  |--- popularity <= 58.50
|  |  |  |  |--- popularity <= 56.50
|  |  |  |  |  |--- class: 56.0
|  |  |  |  |--- popularity >  56.50
|  |  |  |  |  |--- popularity <= 57.50
|  |  |  |  |  |  |--- class: 57.0
|  |  |  |  |  |--- popularity >  57.50
|  |  |  |  |  |  |--- class: 58.0
|  |  |  |--- popularity >  58.50
|  |  |  |  |--- popularity <= 59.50
|  |  |  |  |  |--- class: 59.0
|  |  |  |  |--- popularity >  59.50
|  |  |  |  |  |--- popularity <= 60.50
|  |  |  |  |  |  |--- class: 60.0
|  |  |  |  |  |--- popularity >  60.50
|  |  |  |  |  |  |--- class: 61.0
|  |  |--- popularity >  61.50
|  |  |  |--- popularity <= 66.50
|  |  |  |  |--- popularity <= 63.50
|  |  |  |  |  |--- popularity <= 62.50
|  |  |  |  |  |  |--- class: 62.0
|  |  |  |  |  |--- popularity >  62.50
|  |  |  |  |  |  |--- class: 63.0
|  |  |  |  |--- popularity >  63.50
|  |  |  |  |  |--- popularity <= 64.50
|  |  |  |  |  |  |--- class: 64.0
|  |  |  |  |  |--- popularity >  64.50
|  |  |  |  |  |  |--- popularity <= 65.50
|  |  |  |  |  |  |  |--- class: 65.0
|  |  |  |  |  |  |--- popularity >  65.50
|  |  |  |  |  |  |  |--- class: 66.0
|  |  |  |--- popularity >  66.50
|  |  |  |  |--- popularity <= 70.50
|  |  |  |  |  |--- popularity <= 68.50
|  |  |  |  |  |  |--- popularity <= 67.50
|  |  |  |  |  |  |  |--- class: 67.0
|  |  |  |  |  |  |--- popularity >  67.50
|  |  |  |  |  |  |  |--- class: 68.0
|  |  |  |  |  |--- popularity >  68.50
|  |  |  |  |  |  |--- popularity <= 69.50
|  |  |  |  |  |  |  |--- class: 69.0
|  |  |  |  |  |  |--- popularity >  69.50
|  |  |  |  |  |  |  |--- class: 70.0
|  |  |  |  |--- popularity >  70.50
|  |  |  |  |  |--- popularity <= 74.50
|  |  |  |  |  |  |--- popularity <= 72.50
|  |  |  |  |  |  |  |--- popularity <= 71.50
|  |  |  |  |  |  |  |  |--- class: 71.0
|  |  |  |  |  |  |  |--- popularity >  71.50
```

```
|  |  |  |  |  |  |  |  |--- class: 72.0
|  |  |  |  |  |  |--- popularity >  72.50
|  |  |  |  |  |  |  |--- popularity <= 73.50
|  |  |  |  |  |  |  |  |--- class: 73.0
|  |  |  |  |  |  |  |--- popularity >  73.50
|  |  |  |  |  |  |  |  |--- class: 74.0
|  |  |  |  |  |--- popularity >  74.50
|  |  |  |  |  |  |--- popularity <= 77.50
|  |  |  |  |  |  |  |--- popularity <= 75.50
|  |  |  |  |  |  |  |  |--- class: 75.0
|  |  |  |  |  |  |  |--- popularity >  75.50
|  |  |  |  |  |  |  |  |--- popularity <= 76.50
|  |  |  |  |  |  |  |  |  |--- class: 76.0
|  |  |  |  |  |  |  |  |--- popularity >  76.50
|  |  |  |  |  |  |  |  |  |--- class: 77.0
|  |  |  |  |  |  |--- popularity >  77.50
|  |  |  |  |  |  |  |--- popularity <= 80.50
|  |  |  |  |  |  |  |  |--- popularity <= 78.50
|  |  |  |  |  |  |  |  |  |--- class: 78.0
|  |  |  |  |  |  |  |  |--- popularity >  78.50
|  |  |  |  |  |  |  |  |  |--- popularity <= 79.50
|  |  |  |  |  |  |  |  |  |  |--- class: 79.0
|  |  |  |  |  |  |  |  |  |--- popularity >  79.50
|  |  |  |  |  |  |  |  |  |  |--- class: 80.0
|  |  |  |  |  |  |  |--- popularity >  80.50
|  |  |  |  |  |  |  |  |--- popularity <= 83.50
|  |  |  |  |  |  |  |  |  |--- popularity <= 81.50
|  |  |  |  |  |  |  |  |  |  |--- class: 81.0
|  |  |  |  |  |  |  |  |  |--- popularity >  81.50
|  |  |  |  |  |  |  |  |  |  |--- class: 82.0
|  |  |  |  |  |  |  |  |--- popularity >  83.50
|  |  |  |  |  |  |  |  |  |--- popularity <= 86.50
|  |  |  |  |  |  |  |  |  |  |--- class: 84.0
|  |  |  |  |  |  |  |  |  |--- popularity >  86.50
|  |  |  |  |  |  |  |  |  |  |--- class: 89.0
```

Conclusión:

En el presente trabajo se cumplió con el objetivo de hacer un modelo de machine learning, en el cual se usaron conceptos como arboles de decisión, modelos de aprendizaje e incluso se calculó la precisión del modelo, mismo que tuvo un alto grado de acierto, por lo cual el trabajo fue exitoso.