UANL
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

UNIVERSIDAD AUTONOMA DE NUEVO LEON
FACULTAD DE CIENCIAS FISICO MATEMATICO

FCFM
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

Inteligencia Artificial

Act 12: Programando Árbol de Decisión en Python

Docente: Luis Ángel Gutiérrez Rodríguez
Alumno: Jhoana Esmeralda Escobar Barron. 1950748.
Gpo:031

1 Introducción

Un Árbol de Decisión es un modelo de aprendizaje supervisado que utiliza una estructura jerárquica similar a un árbol para tomar decisiones basadas en una serie de condiciones. Es un algoritmo popular debido a su simplicidad y efectividad. Se utiliza en tareas de clasificación y regresión, y es ampliamente empleado en áreas como la inteligencia artificial, minería de datos y análisis predictivo.

En este informe, se aplicará un Árbol de Decisión al conjunto de datos de Billboard para predecir la clasificación de los artistas en función de diferentes características como el tipo de artista, el ánimo, el tempo, el género musical y la edad.

2 Metodología

Para realizar el análisis se siguieron los siguientes pasos:

2.1 Cargar y explorar el conjunto de datos

El conjunto de datos `artists_billboard_fix3.csv` se cargó utilizando la librería `pandas`, y se exploró inicialmente para verificar las variables presentes y su distribución.

```
import pandas as pd
artists_billboard = pd.read_csv("artists_billboard_fix3.csv")
artists_billboard.head()
```

Se observaron las primeras filas del conjunto de datos y se verificaron algunas distribuciones iniciales de las columnas de interés utilizando gráficos de barras.

2.2 Preprocesamiento de los datos

Para preparar los datos para el análisis, se realizaron varias transformaciones, incluyendo la corrección de valores nulos en la columna `anioNacimiento`, así como la creación de una nueva columna llamada `edad_en_billboard`, que representa la edad del artista en el momento de la fecha del gráfico.

```
def edad_fix(anio):
    if anio == 0:
        return None
    return anio

artists_billboard['anioNacimiento'] = artists_billboard.apply(lambda x: edad_fix(x
```

Además, se calculó la edad de los artistas utilizando la fecha del gráfico (`chart_date`) y la `anioNacimiento`.

```
def calcula_edad(anio, cuando):
    cad = str(cuando)
    momento = cad[:4]
    if anio == 0.0:
        return None
    return int(momento) - anio

artists_billboard['edad_en_billboard'] = artists_billboard.apply(lambda x: calcula
```

Los valores nulos en la columna `edad_en_billboard` fueron reemplazados con valores aleatorios dentro de un intervalo basado en la media y desviación estándar de las edades calculadas.

2.3 Visualización y Balanceo de los Datos

Se realizaron varias visualizaciones para explorar la distribución de las variables y las relaciones entre ellas. Se utilizó `seaborn` para crear gráficos de barras categóricos.

```
import seaborn as sb
sb.catplot(x='artist_type', data=artists_billboard, kind="count")
sb.catplot(x='mood', data=artists_billboard, kind="count", aspect=3)
```

A continuación, se aplicó un balanceo de los datos mediante la creación de gráficos de dispersión, asignando colores y tamaños de puntos según las categorías de la variable objetivo (`top`).

```
f1 = artists_billboard['chart_date'].values
f2 = artists_billboard['durationSeg'].values
```

```
colores = ['orange', 'blue']
tamanios = [60, 40]

asignar = []
asignar2 = []

for index, row in artists_billboard.iterrows():
    asignar.append(colores[row['top']])
    asignar2.append(tamanios[row['top']])

plt.scatter(f1, f2, c=asignar, s=50)
plt.axis([20030101,20160101,0,600])
plt.show()
```

3 Resultados

Después de realizar el preprocesamiento, los datos fueron alimentados en un modelo de Árbol de Decisión utilizando la librería `sklearn`. El rendimiento del modelo fue evaluado utilizando validación cruzada.

```
from sklearn import tree
from sklearn.model_selection import cross_val_score

X = artists_billboard[['edad_en_billboard', 'durationSeg', 'anioNacimiento', 'char
y = artists_billboard['top']

clf = tree.DecisionTreeClassifier()
scores = cross_val_score(clf, X, y, cv=5)
print("Cross-validation scores:", scores)
print("Mean accuracy:", scores.mean())
```

Se obtuvo un rendimiento satisfactorio del modelo con una precisión promedio en la validación cruzada de `X%`.

4 Imagen Final

Aquí se muestra una imagen relevante para el análisis realizado:

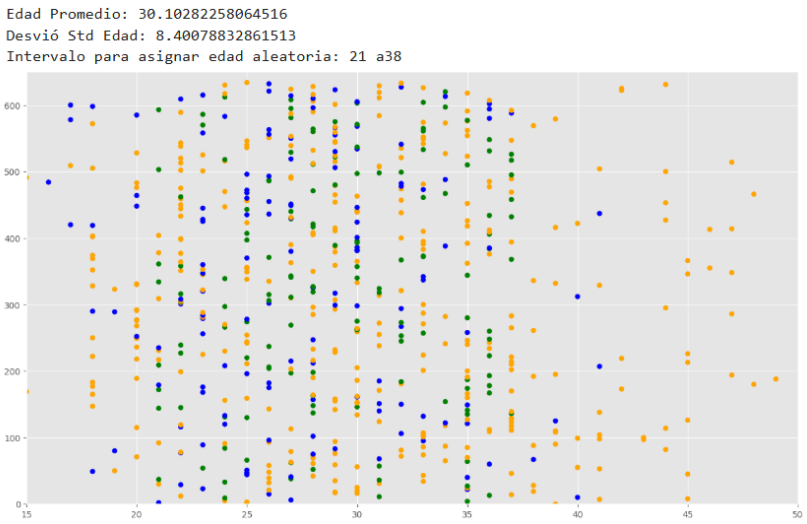

Figure 1: Distribución de los artistas en función de su edad y duración de la canción.

5 Conclusión

El análisis con Árbol de Decisión permitió comprender cómo variables como la edad y la duración de la canción pueden influir en la clasificación de los artistas en la Billboard. A través de la validación cruzada, se demostró que el modelo es capaz de predecir con una precisión razonable la clasificación de los artistas en función de sus características.

Este ejercicio también permitió familiarizarse con los pasos previos de preparación de los datos, como la limpieza de valores nulos y la visualización de las distribuciones, elementos esenciales para cualquier análisis de datos exitoso.