UNIVERSIDAD AMERICANA
Facultad de Ingeniería y Arquitectura

Inteligencia de negocios

Preprocesamiento y transformación de datos

Estudiante:
Carlos Diego Toruño Espinales

Docente:
Arlen Jeannette Lopez

Fecha:

16 de Septiembre 2024

Introducción

El preprocesamiento de datos es un paso fundamental en cualquier análisis de datos. A través de técnicas como la imputación de valores faltantes, la detección y tratamiento de outliers, y la normalización y estandarización de variables, se busca mejorar la calidad de los datos y prepararlos para un análisis adecuado.

Limpieza de datos

- **Exploración inicial del conjunto de datos**

El primer paso del análisis consiste en una exploración inicial del conjunto de datos, en la cual se revisan las primeras filas para comprender la estructura general de la información. Esto incluye identificar el tipo y número de variables presentes, así como obtener una descripción básica de cada una para evaluar cómo deben ser procesadas y transformadas.

```
url = 'https://raw.githubusercontent.com/cdtoruno/new-repo/main/Spotify_Youtube.csv'
data = pd.read_csv(url)

# Exploracion inicial del dataset
print(data.head())
print(data.shape)
print(data.info())
print(data.describe())
```

- **Identificación de variables relevantes**

Dada la cantidad de columnas en el conjunto de datos, es necesario reducir las opciones para enfocar el análisis de manera más precisa y relevante. Para este propósito, se han seleccionado las siguientes columnas clave para la exploración y el análisis.

```
# Seleccionar las columnas importantes
columnas_importantes = ['Energy', 'Views', 'Likes',
 'Comments', 'Stream', 'Licensed', 'official_video']
```

Con las columnas clave ya identificadas, se realiza un análisis de las variables que contienen valores faltantes, presentándose en forma de porcentaje. Esto facilita la visualización y comprensión de la magnitud de los datos faltantes en relación con el conjunto total.

```
# Identificar valores faltantes
print('\nValores faltantes por columna:')
missing_values = data.isnull().sum()
print(missing_values[missing_values > 0])
```

- **Manejo de valores faltantes**

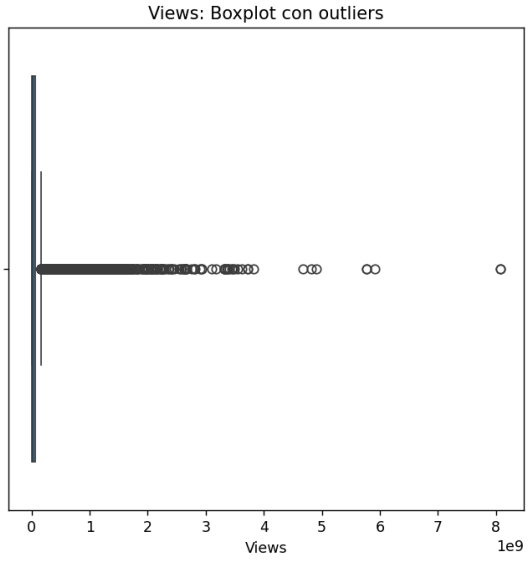
Para tratar los valores faltantes, se optaron por dos enfoques clave. Para las variables numéricas, se utilizó la mediana, ya que es menos afectada por la presencia de outliers y proporciona una representación más estable de los datos centrales. En cuanto a las variables categóricas, se aplicó la moda, que reemplaza los valores faltantes con la categoría más frecuente. Esta combinación de técnicas asegura que los datos sean imputados de manera adecuada, minimizando el impacto de valores atípicos y respetando la naturaleza categórica de ciertos atributos.

```
# Imputar valores faltantes con la mediana para las columnas numéricas
for column in ['Energy', 'Views', 'Likes', 'Comments', 'Stream']:
    datos_importantes.loc[:, column] = datos_importantes.loc[:, column].fillna(datos_importantes[column].median())

# Imputar valores faltantes con la moda para las columnas categóricas
for column in ['Licensed', 'official_video']:
    datos_importantes.loc[:, column] = datos_importantes.loc[:, column].fillna(datos_importantes[column].mode()[0])
```

- **Detección de valores atípicos**

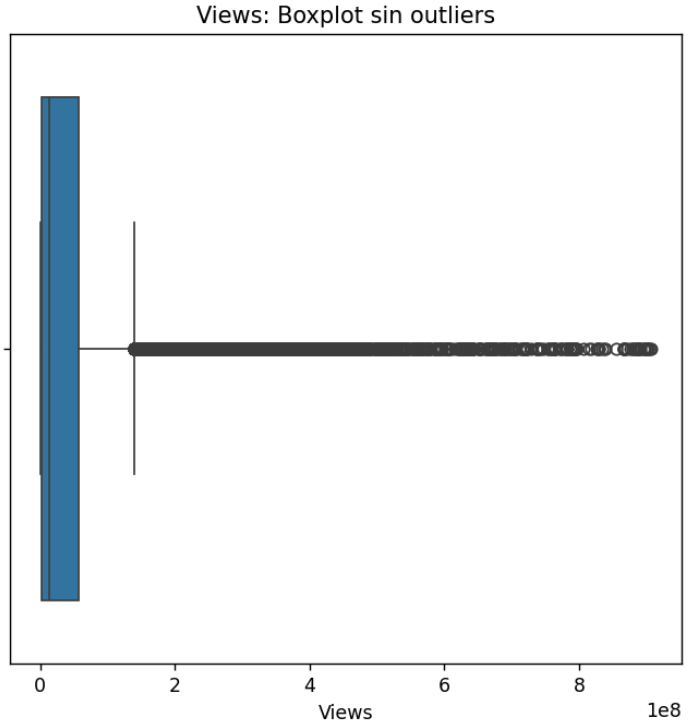
Después de imputar los valores faltantes, surge otra preocupación importante: los valores atípicos. Estos valores representan puntos numéricos significativamente alejados del resto del conjunto de datos y pueden afectar negativamente el análisis. Para identificarlos y gestionarlos adecuadamente, recurrimos a gráficos visuales, como los boxplots, que nos permiten observar de manera clara cualquier inconsistencia o comportamiento inusual en los datos, ayudando a tomar decisiones informadas sobre su tratamiento.



En la gráfica podemos observar que algunas columnas presentan valores que están significativamente alejados del resto de los datos. Para eliminar estos valores atípicos, utilizamos la técnica Z-Score, que mide cuántas desviaciones estándar están un valor por encima o por debajo de la media de los datos numéricos. En este caso, hemos definido un umbral de 3, lo que significa que cualquier valor con una desviación estándar superior a 3 o inferior a -3 se considera atípico y es tratado en consecuencia. Esto nos permite mantener solo los datos que se encuentran dentro de un rango más cercano a la media, mejorando la calidad del análisis posterior.

```
# Aplicar Z-score para detección de outliers en las columnas numéricas
z_scores = np.abs(stats.zscore(datos_importantes[['Energy', 'Views', 'Likes', 'Comments', 'Stream']]))
threshold = 3
outliers_condition = (z_scores < threshold).all(axis=1)
```

Gráfico después del manejo de los outliers



- **Transformación de variables**

Otra forma de manejar variables es mediante la normalización (Min-Max Scaling), que ajusta los datos a un rango específico para hacerlos más comparables, ideal para modelos de machine learning. A diferencia de Z-Score, que puede generar valores negativos y no tiene un rango fijo, Min-Max Scaling garantiza que los datos estén dentro de un rango predefinido. Para implementarla, utilizamos la librería sklearn.preprocessing.

```
# Normalización (Min-Max Scaling) para las columnas numéricas
scaler_minmax = MinMaxScaler()
minmax_scaled_data = scaler_minmax.fit_transform(datos_sin_outliers[['Energy', 'Views', 'Likes', 'Comments', 'Stream']])
minmax_scaled_df = pd.DataFrame(minmax_scaled_data, columns=['Energy', 'Views', 'Likes', 'Comments', 'Stream'])

# Estandarización (Z-score Scaling) para las columnas numéricas
scaler_standard = StandardScaler()
zscore_scaled_data = scaler_standard.fit_transform(datos_sin_outliers[['Energy', 'Views', 'Likes', 'Comments', 'Stream']])
zscore_scaled_df = pd.DataFrame(zscore_scaled_data, columns=['Energy', 'Views', 'Likes', 'Comments', 'Stream'])
```

A continuación se muestra un ejemplo de la diferencia entre estas dos técnicas en el conjunto de datos:

Min-Max

	Energy	Views	Likes	Comments	Stream
0	0.702994	0.079425	0.181875	0.052348	0.361983
1	0.922998	0.009303	0.047552	0.012493	0.073612
2	0.738995	0.233553	0.301445	0.093254	0.507416
3	0.890998	0.285685	0.310896	0.121585	0.378054
4	0.896998	0.000499	0.001970	0.000407	0.012444

Z-Score

	Energy	Views	Likes	Comments	Stream
0	0.320426	0.127373	0.887292	0.522568	1.507964
1	1.341206	-0.434479	-0.174322	-0.187013	-0.266634
2	0.487463	1.362342	1.832304	1.250848	2.402948
3	1.192729	1.780053	1.907006	1.755256	1.606864
4	1.220569	-0.505028	-0.534580	-0.402196	-0.643058

- **Ingeniería de características**

A partir de las variables existentes en nuestro conjunto de datos, podemos crear nuevas variables combinando o transformando las ya disponibles. En este caso, hemos generado dos nuevas variables relevantes para nuestro análisis. La primera es Engagement_Rate, que se calcula como la proporción de Likes y Comments sobre el total de Views, lo que nos da una idea del nivel de interacción en relación con la visibilidad del contenido.

La segunda variable, Interaction_Score, mide la relación entre Likes y Comments sobre el total de Streams, lo que nos permite evaluar el nivel de interacción en función del número de veces que se ha escuchado una canción o visto un video. Estas nuevas variables enriquecen el análisis al proporcionar métricas adicionales de comportamiento e interacción.

```
# Creacion de nuevas variables
# Proporción de Likes y Comments sobre Views
datos_sin_outliers['Engagement_Rate'] = (datos_sin_outliers['Likes'] + datos_sin_outliers['Comments']) / datos_sin_outliers['Views']

# Proporción de Likes y Comments sobre Stream
datos_sin_outliers['Interaction_Score'] = (datos_sin_outliers['Likes'] + datos_sin_outliers['Comments']) / datos_sin_outliers['Stream']
```

- **Codificación de variables categóricas**

La codificación de variables es el proceso de transformar variables categóricas (es decir, aquellas que contienen categorías o etiquetas, como "sí" o "no") en un formato numérico que pueda ser interpretado por algoritmos de aprendizaje automático. En este caso, para el conjunto de datos actual, hemos aplicado **One-Hot Encoding** a las variables categóricas Licensed y official video. Esta técnica transforma cada categoría en una nueva columna binaria, donde se asigna un valor de 1 si la categoría está presente o 0 si no lo está.

Codificación de variables categóricas

	Energy	Views	Likes	Comments	Stream	Engagement_Rate	Interaction_Score	Licensed_True	official_video_True
1	0.703	72011645.0	1079128.0	31003.0	310083733.0	0.015416	0.003580	True	True
2	0.923	8435055.0	282142.0	7399.0	63063467.0	0.034326	0.004591	True	True
3	0.739	211754952.0	1788577.0	55229.0	434663559.0	0.008707	0.004242	True	True
5	0.891	259021161.0	1844658.0	72008.0	323850327.0	0.007400	0.005918	True	True
6	0.897	451996.0	11686.0	241.0	10666154.0	0.026387	0.001118	False	True

- **Comparación antes y después del conjunto de datos**

Estado del conjunto de datos antes:

Antes de aplicar el preprocesamiento, el conjunto de datos contenía diversas inconsistencias que complicaba su análisis. Muchas de las variables exhiben distribuciones sesgadas con la presencia de valores atípicos que afectan la interpretación general de los datos. Además, se observaban valores faltantes en varias columnas, lo que incrementa la posibilidad de obtener resultados poco precisos o no representativos. Estos problemas requerían una adecuada limpieza y transformación para garantizar un análisis más robusto y confiable.

Antes del preprocesamiento:

	Unnamed: 0	Danceability	Energy	Key	Loudness	...	Duration_ms	Views	Likes	Comments	Stream
count	20718.000000	20716.000000	20716.000000	20716.000000	20716.000000	...	2.071600e+04	2.024800e+04	2.017700e+04	2.014900e+04	2.014200e+04
mean	10358.500000	0.619777	0.635250	5.300348	-7.671680	...	2.247176e+05	9.393782e+07	6.633411e+05	2.751899e+04	1.359422e+08
std	5980.915774	0.165272	0.214147	3.576449	4.632749	...	1.247905e+05	2.746443e+08	1.789324e+06	1.932347e+05	2.441321e+08
min	0.000000	0.000000	0.000020	0.000000	-46.251000	...	3.098500e+04	0.000000e+00	0.000000e+00	0.000000e+00	6.574000e+03
50%	10358.500000	0.637000	0.666000	5.000000	-6.536000	...	2.132845e+05	1.450110e+07	1.244810e+05	3.277000e+03	4.968298e+07
75%	15537.750000	0.740250	0.798000	8.000000	-4.931000	...	2.524430e+05	7.039975e+07	5.221480e+05	1.436000e+04	1.383581e+08
max	20717.000000	0.975000	1.000000	11.000000	0.920000	...	4.676058e+06	8.079649e+09	5.078865e+07	1.608314e+07	3.386520e+09

Estado del conjunto de datos después:

Tras el preprocesamiento, el conjunto de datos mostró una mejora significativa en su estructura y calidad. La eliminación de los valores atípicos y la aplicación de técnicas de normalización contribuyeron a una representación más equilibrada y consistente de las distribuciones. Además, la imputación de valores faltantes permitió preservar información clave, asegurando una base más sólida y confiable para un análisis posterior más preciso y exhaustivo.

Después del preprocesamiento:

	Energy	Views	Likes	Comments	Stream	Engagement_Rate	Interaction_Score
count	19943.000000	1.994300e+04	1.994300e+04	19943.000000	1.994300e+04	19942.000000	19943.000000
mean	0.633941	5.759866e+07	4.130111e+05	13619.921376	1.001784e+08	0.026562	0.021069
std	0.215527	1.131582e+08	7.507491e+05	33265.535234	1.392013e+08	0.946706	1.284357
min	0.000020	0.000000e+00	0.000000e+00	0.000000	6.574000e+03	0.000000	0.000000
25%	0.505000	1.757280e+06	2.114100e+04	503.000000	1.745295e+07	0.005967	0.000810
50%	0.665000	1.391414e+07	1.217060e+05	3203.000000	4.893321e+07	0.008953	0.002731
75%	0.798000	5.705871e+07	4.292230e+05	11727.000000	1.186494e+08	0.015479	0.006746
max	1.000000	9.066678e+08	5.933354e+06	592245.000000	8.566147e+08	91.397944	172.462997