

INTELIGENCIA ARTIFICIAL Y MACHINE LEARNING

DATOS VACIOS

ANDRES ANTONIO CERVANTES CONSUEGRA

LUIS FERNANDO PICON CARRILLO

CAMILA ANDREA ROSALES MERCADO

JORGE RUDAS

TECNOLOGÍA EN GESTIÓN DE SISTEMAS INFORMÁTICOS

INSTITUCIÓN DE EDUCACIÓN SUPERIOR HUMBERTO VELASQUE GARCIA
(INFOTEP)

CIENAGA - MAGDALENA

2024

introducción: verificamos y analizamos los datos null(vacios) para definir la calidad de información si no hay datos suministrados para hacer una estrategia para definir si dichos datos pueden ser suministrados o eliminados por un algoritmo.

Objetivos:

identificar si el dato es (null) vacío se elimina

Analizar si la cantidad de datos suministrada es deficiente

detectar y procesar los datos de manera efectiva para garantizar que el conjunto de datos, sea utilizado de manera uniforme.

Defina una métrica de calidad de datos vacíos

Es la medida utilizada para evaluar el impacto de los valores faltantes en un conjunto de datos. Ayuda a determinar la calidad de los datos en un análisis y en tomas de decisiones lo podemos encontrar en frecuencias de valores faltantes por columnas, distribución de valores faltantes, impacto en la completitud del dataset y en la relación con otras variables.

Utilizar visualizaciones para identificar columnas con valores faltante, incluyen valores NaN (no un número) o NULOS.

Determine el impacto de los valores que faltan en su análisis o modelo. Tener en cuenta el porcentaje de valores que faltan en cada columna y su importancia para el conjunto de datos global.

Para las funciones numéricas, puede imputar los valores que faltan usando técnicas como la media, la mediana o el método de imputación de modo (fillna() en pandas). Para las características categóricas, puede imputar con la categoría más frecuente.

Defina un algoritmo para evaluar la calidad de los datos

1. para la revisión de los valores vacíos lo que podemos hacer agregar la función isnull para decir que hacen falta datos en la columna

```
print(df.isnull().sum()) # Cuenta valores faltantes por columna
```

2. Visualizar los datos faltantes

Podemos visualizar los datos por medio de un gráfico los gráficos recién faltantes

```
msno.matrix(df)
```

También puedes usar:

4. Evaluar la completitud del dataset

Aquí podemos ver o analizar que tan completo estan los datos en un 100%

```
completitud = df.dropna().shape[0] / df.shape[0] * 100
```

```
print(f"Porcentaje de registros completos: {completitud:.2f}%")
```

5. Detectar patrones en los datos vacíos

Identificar si los datos faltantes son aleatorios o tienen patrones

```
df['flag_missing'] = df.isnull().sum(axis=1) # Número de valores vacíos por fila
```

```
print(df.groupby('flag_missing').size()) # Ver cuántas filas tienen cada cantidad de valores
```

vacíos

6. Aplicar imputación de valores (Opcional)

Defina un algoritmo para procesar sus datos

Nosotros tomamos la decision de eliminar los datos con las celdas y columnas que no nos muestre datos y tengan una relacion enntre ellas, sino modificar los datos según necesario

```
# Eliminar filas con valores vacíos
```

```
df.dropna(inplace=True)
```

```
# Convertir texto a minúsculas
```

```
df['Habitaciones'] = df[' Habitaciones '].str.lower()
```

```
# Crear una nueva columna con un cálculo
```

```
df['Estado'] = df['Estado'] * 5
```

```
# Normalizar una columna numérica
```

```
df[precio] = (df['precio'] - df['precio'].min()) / (df['precio'].max() - df['precio'].min())
```

conclusión: según los datos nulos (null) en SQL son importantes en el proceso de análisis de datos, debido a saber las métricas requeridas para saber si el archivo es de alta prioridad o visualizar el problema que está afectando la base y que decisión tomar y como solucionarlo.