Francisco Javier Molina Rojas

Generación de Datos sintéticos usando SDV (Synthetic Data Vault)

- **Introducción**

Synthetic Data Vault (SDV más Adelante), es una librería de Python la cual contiene un ecosistema de librerías que nos permiten generar datos sintéticos. Estos datos son distintos a los originales, pero guardan el mismo formato y propiedades estadísticas.

Desde el punto de vista de la protección de datos, los datos sintéticos, son datos que no pertenecen a ningún individuo, por lo que no es necesaria su anonimización.

- **Como usar SDV**

SDV puede ser sencillamente instalado en su sistema usando los siguientes comandos:

- Pip: **pip install sdv**
- Conda: **conda install -c pytorch -c conda-forge sdv**

- **Ejemplo de uso**

Para este ejemplo usaremos un dataset disponible en la librería sklearn. En este caso usaremos el dataset llamado: “The Iris Dataset”. Este contiene mediciones de longitud/ancho de los sépalos/pétalos de ciertas flores (Setosa, Versicolor y Virginica).

Una muestra de 4 ejemplares de este dataset podría ser la siguiente:

sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
5.1	3.5	1.4	0.2
4.9	3.0	1.4	0.2
4.7	3.2	1.3	0.2
4.6	3.1	1.5	0.2

Para poder obtener datos sintéticos de este dataset usaremos el siguiente algoritmo:

```python
import pandas as pd
# Libreria para la manipulacion de datos
from sklearn.datasets import load_iris
# Usaremos el dataset iris
from sdv.tabular import CTGAN
# Tecnica de sintesis de datos
import matplotlib.pyplot as plt
# Libreria para poder realizar la regresionç


# Cargamos el dataset de iris de scikit-learn
iris = load_iris()

# Usamos los datos del dataset como una estructura dat
a frame gracias a panda
original_data = pd.DataFrame(data=iris.data, columns
=iris.feature_names)

# Print del dataset original
print("Dataset original:\n")
print(original_data)

# Seleccionar las columnas a sintetizar
columns_to_synthesize = ['sepal length (cm)',
'sepal width (cm)', 'petal length (cm)',
'petal width (cm)']

# Seleccionar el modelo de machine o deep learning a u
sar (por ejemplo, CTGAN)
model = CTGAN()

# Entrenar el modelo con los datos originales
model.fit(original_data[columns_to_synthesize])

# Generar nuevos datos sintéticos
synthetic_data = model.sample(len(original_data))

# Mostrar el dataset generado
print("Dataset generado:\n")
print(synthetic_data)
```

Figura 1: example.py

Nótese el uso de otras librerías como pandas (para usar estructura de datos), sklearn (para obtener el dataset) y matplotlib (para la generación de los resultados).

El modelo que usamos es conocido como CTGAN, es un modelo de Deep Learning constituido por una red neuronal.

Además de CTGAN, SDV nos permite usar otros modelos.

- **Estudio de los resultados**

Vamos a realizar un estudio de los datos. Para ello veremos la distribución de los datos originales y generados según la longitud y el ancho de los sépalos.

Para esto usaremos el siguiente algoritmo:

```
# Ahora vamos a realizar un analisis de regresion de la
longitud y ancho de los sepalos, para ver la calidad de
los datos generados

# Para generar los puntos
plt.scatter(original_data['sepal length (cm)'
], original_data['sepal width (cm)'], label=
'Datos originales', c ='blue')

# aplicar leyenda
plt.xlabel('sepal length (cm)')
plt.ylabel('sepal width (cm)')
plt.legend()

#mostrar grafico
plt.show()
```

Figura 2: Código Regresión

Obtenemos los siguientes resultados:

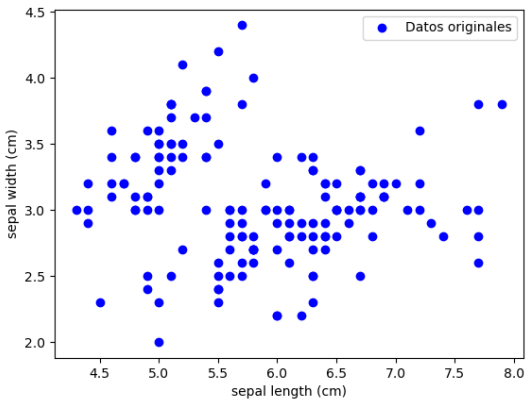

Figura 3: Datos originales

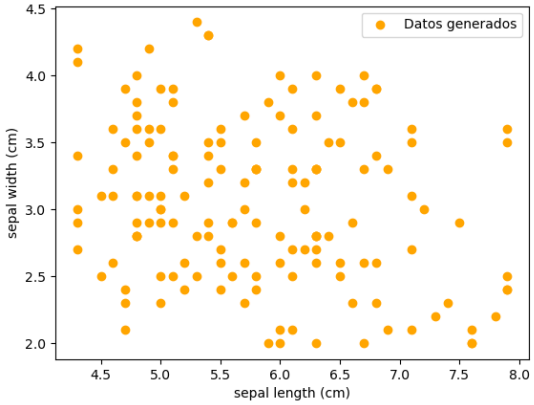

Figura 4: Datos generados

Como podemos ver, los resultados no son los mejores posibles. Esto se puede deber, además del tipo de modelo usado, tambien por el entrenamiento que hemos realizado. Este puede ser ampliado, lo que dará con el tiempo mejores resultados.