TALENTO TECH

TIC

Universidad Tecnológica de Bolívar

Exploratory Data Analysis 1

Ejecutor técnico: Jorge Luis Villalba Acevedo

www.utb.edu.co/talento-tech

Librerias

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import
MinMaxScaler, StandardScaler
```

Normalización y Escalado de Datos

Normalización

- Es el proceso de transformar datos a un rango común, generalmente [0, 1]. Se utiliza cuando los datos no siguen una distribución normal, pero las características deben estar en el mismo rango para ciertos algoritmos, como redes neuronales o métodos basados en distancias.

Matemáticamente:

- **Normalización Min-Max**:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Donde:

- x es el valor original.
- $min(x)$ es el valor mínimo.
- $max(x)$ es el valor máximo.

Ejemplo 1.

- Utilizando 10 valores aleatorios que no siguen una distribución normal y responde los siguientes interrogantes.

```
# Generar 10 valores aleatorios que no siguen una distribución normal

# Definir una semilla
np.random.seed(42)

data = np.random.randint(low=10, high=100, size=10)
df = pd.DataFrame(data, columns=['Valores'])

print(data)
```

```
[61 24 81 70 30 92 96 84 84 97]
```

1. ¿Cuál es el valor mínimo del conjunto de datos?
2. ¿Cuál es el valor máximo del conjunto de datos?
3. ¿Cómo se distribuyen los datos?
4. Crear un gráfico de dispersión

Ejemplo 2.

- Normaliza los observaciones y responde los siguientes interrogantes.

1. **¿Cuál es el valor mínimo del conjunto de datos?**
2. **¿Cuál es el valor máximo del conjunto de datos?**
3. **¿Cómo se distribuyen los datos?**
4. **Crear un gráfico de dispersión**

Normalización características

- Rango de valores: [0, 1].
- Utilizado cuando los datos no tienen una distribución gaussiana.
- Sensible a outliers.

En python

```
# Normalización (Min-Max Scaling)
scaler_min_max = MinMaxScaler()
scaler_min_max.fit_transform(df)
```

```
array([[0.50684932],
       [0.        ],
       [0.78082192],
       [0.63013699],
       [0.08219178],
       [0.93150685],
       [0.98630137],
       [0.82191781],
       [0.82191781],
       [1.        ]])
```

Escalado (Standardization)

- Transforma los datos para que tengan media cero y desviación estándar uno. Es útil cuando los datos siguen una distribución normal o cuando se usan modelos que asumen dicha distribución, como regresión lineal o métodos basados en SVM.

Estandarización

$$z = \frac{x - \mu}{\sigma}$$

Donde:

- x es el valor original.
- μ es la media de los datos.
- σ es la desviación estándar de los datos.

Ejemplo 3

- Utilizando los mismo datos del ejemplo del 1, responde las preguntas siguientes

1. **¿Cuál es la media del conjunto de datos?**
2. **¿Cuál es la desviación estándar del conjunto de datos?**
3. **¿Cómo se distribuyen los datos?**
4. **Crear un gráfico de dispersión**

Ejemplo

- Estandariza las observaciones y responde los siguientes interrogantes.

1. **¿Cuál es la media del conjunto de datos estandarizados?**
2. **¿Cuál es la desviación estándar del conjunto de datos estandarizados?**
3. **¿Cómo se distribuyen los datos estandarizados?**
4. **Crear un gráfico de dispersión**

Estandarización características

- Media = 0, Desviación estándar = 1.
- Adecuado cuando los datos tienen distribución gaussiana.
- Menos sensible a outliers que la normalización.

En python

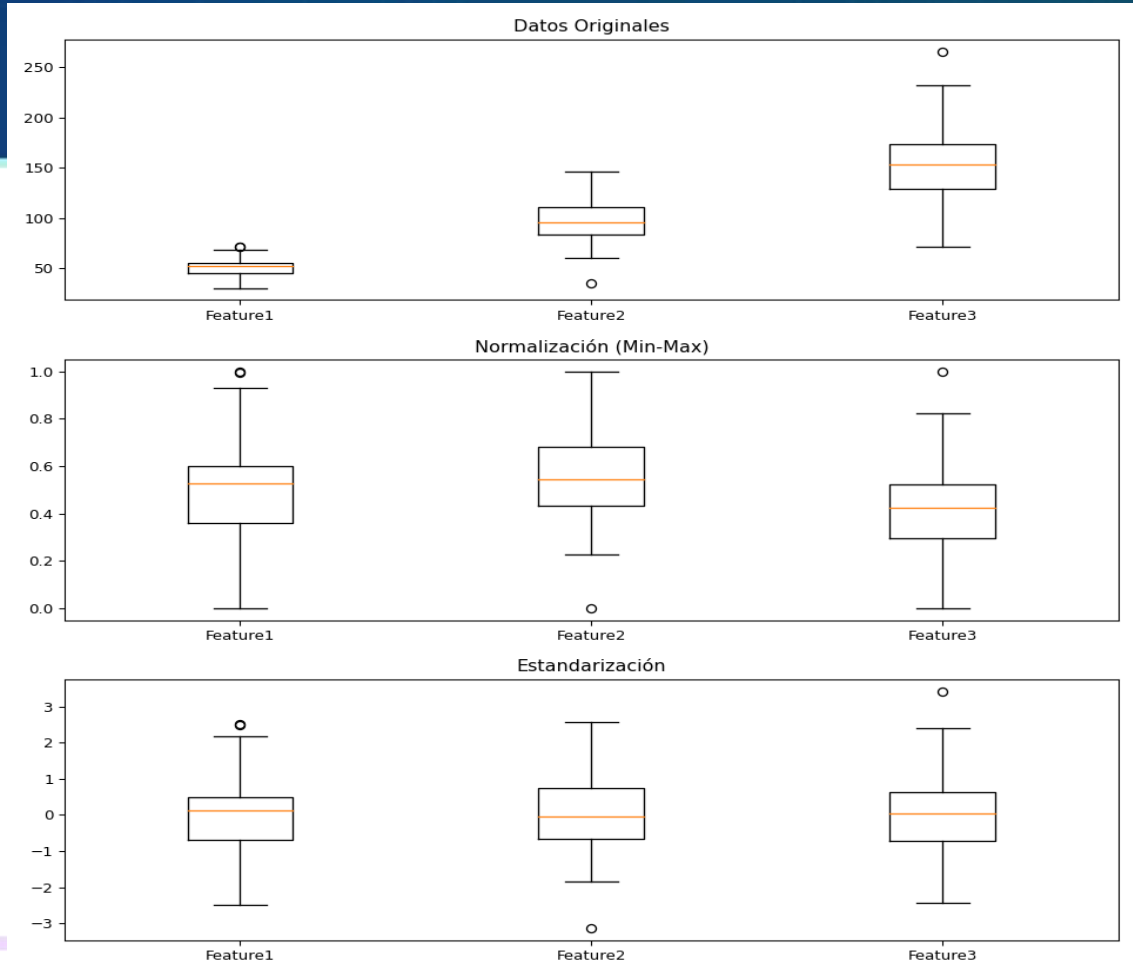
```
# Estandarización (Standardization)
scaler_standard = StandardScaler()
scaler_standard.fit_transform(df)
```

```
array([[-0.43907017],
       [-1.92949187],
       [ 0.36656317],
       [-0.07653517],
       [-1.68780187],
       [ 0.80966152],
       [ 0.97078819],
       [ 0.48740818],
       [ 0.48740818],
       [ 1.01106985]])
```

Implementación en Python con Datos Simulados

```
# Simulación de datos
np.random.seed(42)
data = np.random.randn(100, 3) * [10, 20, 30]
+ [50, 100, 150]  # Datos con diferentes
escalas
df = pd.DataFrame(data, columns=['Feature1',
'Feature2', 'Feature3'])
#df
```

Ejemplo


Datos Originales
250
200
150
100
50
Feature1
Feature2
Feature3
Normalización (Min-Max)
1.0
0.8
0.6
0.4
0.2
0.0
Feature1
Feature2
Feature3
Estandarización
3
2
1
0
−1
−2
−3
Feature1
Feature2
Feature3

Referencias

1. **McKinney, W. (2017).** *Python for data analysis: Data wrangling with pandas, numpy, and ipython* (2nd ed.). O'Reilly Media.
2. **Sweigart, A. (2019).** *Automate the boring stuff with Python: Practical programming for total beginners* (2nd ed.).
3. **González Duque, R. (2015).** *Python para todos*. Independently published.
4. **Bosch, J. (2020).** *Introducción a la programación en Python: Algoritmos y lógica para principiantes*. Independently published.