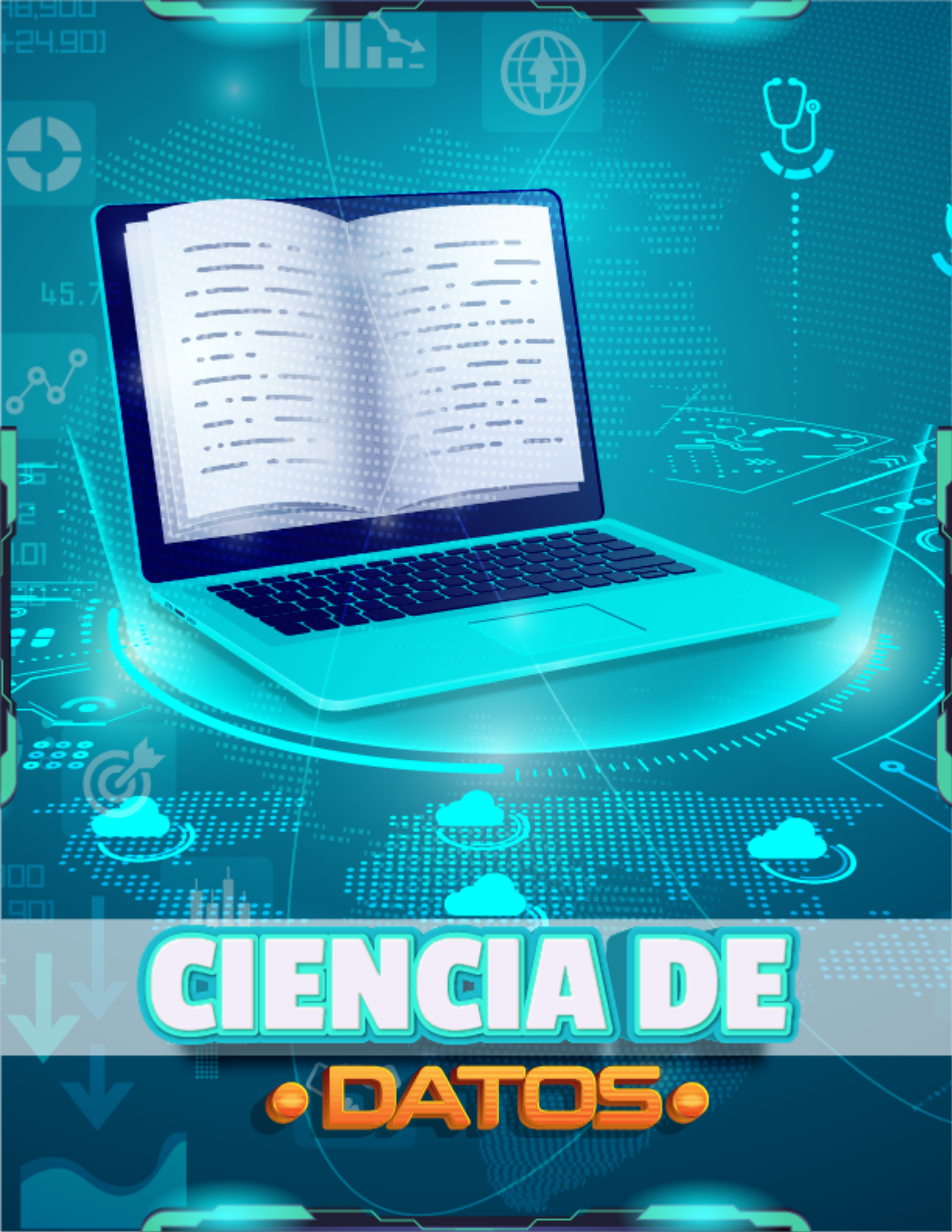
CIENCIA DE
DATOS

Guía de Comandos para el Procesamiento de Datos en Python

Introducción

El lenguaje de programación Python ha crecido en popularidad en los últimos años debido a su simplicidad y eficacia, especialmente en el ámbito del análisis de datos. Las bibliotecas de Python, como pandas, numpy y sklearn, proporcionan un conjunto robusto y completo de herramientas para el procesamiento y análisis de datos. Esta guía se centra en los comandos más comúnmente utilizados en Python para eliminar duplicados, analizar datos faltantes, estandarizar datos, agregar datos y otros pasos clave en el procesamiento de datos.

Eliminación de Duplicados

drop_duplicates()

La función drop_duplicates() de la biblioteca pandas se utiliza para eliminar duplicados de un DataFrame. Por defecto, considera todas las columnas.
import pandas as pd

```
# Crear DataFrame
df = pd.DataFrame({'A': ['foo', 'bar', 'foo', 'bar'], 'B': ['one', 'one', 'two', 'two'], 'C': [1, 2, 2, 3]})
# Eliminar duplicados
df = df.drop_duplicates()
```

Se puede especificar un subconjunto de columnas para considerar en la detección de duplicados.

```
df = df.drop_duplicates(subset=['A', 'B'])
```

Recomendación: Es recomendable realizar la eliminación de duplicados después de la limpieza de datos para evitar la eliminación incorrecta de datos.

Análisis de Datos Faltantes

isnull(), notnull()

Las funciones isnull() y notnull() de pandas se utilizan para detectar valores faltantes. Devuelven un DataFrame de Booleanos que es True donde los valores están ausentes o False donde no lo están.

```
# Detectar valores nulos
nulls = df.isnull()
# Detectar valores no nulos
non_nulls = df.notnull()
dropna()
```

La función dropna() se utiliza para eliminar filas y/o columnas con valores nulos.

```
# Eliminar filas con valores nulos
df = df.dropna()
# Eliminar columnas con valores nulos
df = df.dropna(axis=1)
```

Recomendación: El uso de dropna() debe hacerse con cuidado, ya que puede resultar en la pérdida de datos útiles. A veces, es mejor imputar los valores faltantes en lugar de eliminarlos.

Estandarización de Datos

StandardScaler()

La clase StandardScaler de la biblioteca sklearn se utiliza para estandarizar los datos al eliminar la media y escalar a la varianza de la unidad.

from sklearn.preprocessing
import StandardScaler

```
# Crear objeto
StandardScaler scaler = StandardScaler()
# Ajustar y transformar los datos
df_scaled = scaler.fit_transform(df)
```

Recomendación: La estandarización es una técnica importante para muchas técnicas de aprendizaje automático, ya que pueden ser sensibles a la escala de las características.

Agregación de Datos

groupby()

La función groupby() de pandas se utiliza para agrupar los datos por ciertas características y aplicar funciones a cada grupo.

```
# Agrupar por columna 'A' y calcular la media de
'C'
df_grouped = df.groupby('A')['C'].mean()
pivot_table()
```

La función pivot_table() de pandas proporciona una forma de resumir y agregar datos de manera similar a las tablas dinámicas de Excel.

```
# Crear tabla pivotante
pivot = pd.pivot_table(df, values='C', index='A',
columns='B', aggfunc=np.sum)
```

Recomendación: El uso de la agregación de datos puede ayudar a resumir y entender grandes conjuntos de datos, pero es importante considerar qué tipo de agregación es apropiada para cada situación.

Otros Pasos del Procesamiento de Datos

replace()

La función replace() de pandas permite reemplazar un valor por otro.
pythonCopy code

```
# Reemplazar
'foo' por 'baz' df = df.replace('foo', 'baz')
merge()
```

La función merge() se utiliza para combinar dos DataFrames en uno solo.

```
# Unir dos DataFrames
df = df1.merge(df2, on='key')
```

Recomendación: Es importante entender bien las implicaciones de las operaciones de fusión, ya que pueden cambiar drásticamente la estructura de los datos.

Esta guía proporciona un resumen de los comandos más comúnmente utilizados en Python para el procesamiento de datos. Sin embargo, el procesamiento de datos es un campo amplio y cada conjunto de datos puede requerir un enfoque diferente. Por lo tanto, es importante entender bien las herramientas a su disposición y cómo se pueden aplicar a sus propios datos.