



Módulo 1 - Unidad 2 - Lección 2

Actividad 2

Pandas

¿Por qué Pandas es esencial para el análisis de datos y manipulación de datos en IA?

Pandas es esencial para el análisis de datos y la manipulación de datos en inteligencia artificial por varias razones fundamentales como:

Estructuras de Datos Poderosas:

Pandas ofrece dos estructuras de datos principales: series y DataFrames. Los DataFrames son especialmente poderosos y permiten organizar datos de manera tabular, similar a una hoja de cálculo, con filas y columnas, estas estructuras son flexibles y eficientes para representar y manipular datos complejos.



¿Por qué Pandas es esencial para el análisis de datos y manipulación de datos en IA?

Manipulación de Datos Eficiente:

Pandas proporciona una amplia gama de funciones para la manipulación de datos, como filtrado, selección, agregación y transformación. Estas operaciones son esenciales en el preprocesamiento de datos para el entrenamiento de modelos de aprendizaje automático.

Integración con NumPy:

Pandas se integra bien con NumPy, otra biblioteca esencial en el ámbito de la inteligencia artificial. Los DataFrames de Pandas pueden contener columnas de tipo NumPy array, lo que facilita la integración de las funcionalidades de ambas bibliotecas.



¿Por qué Pandas es esencial para el análisis de datos y manipulación de datos en IA?

Manejo de Datos Faltantes:

El mundo real a menudo presenta conjuntos de datos con valores faltantes o nulos. Pandas ofrece herramientas para manejar estos casos, ya sea eliminando datos faltantes o llenándolos con valores específicos.

Indexación Intuitiva:

- La indexación en Pandas permite acceder y manipular datos de manera intuitiva, pueden indexar DataFrames por etiquetas de fila y columna, facilitando la referencia a datos específicos y la realización de operaciones complejas.



¿Por qué Pandas es esencial para el análisis de datos y manipulación de datos en IA?

Operaciones de Agrupación y Agregación:

Pandas facilita la realización de operaciones de agrupación y agregación de datos, pueden agrupar datos en función de ciertos criterios y aplicar funciones de agregación, como sumas o promedios a grupos específicos.

Entrada y Salida de Datos Simplificada:

Pandas simplifica la entrada y salida de datos en diversos formatos como CSV, Excel, SQL y más. Esto facilita la carga y la escritura de datos desde y hacia diferentes fuentes, lo cual es crucial en el ciclo de vida de un proyecto de inteligencia artificial.



Pandas



¿Por qué Pandas es esencial para el análisis de datos y manipulación de datos en IA?

Visualización de Datos Integrada:

Aunque no es la función principal de Pandas, la biblioteca proporciona herramientas para visualizar datos de manera rápida y sencilla, lo cual es útil en la exploración inicial de datos.



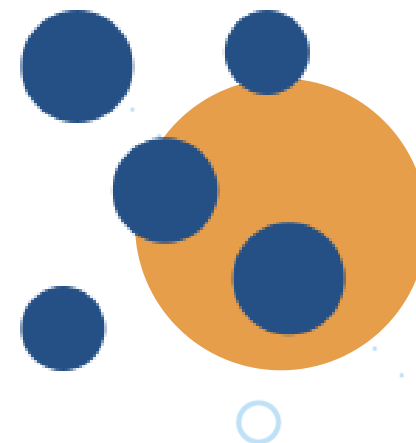
Presentar Series y DataFrames como estructuras principales de Datos en Pandas

En Pandas, las estructuras de datos fundamentales son las Series y los DataFrames.

Series:

Una Serie es un arreglo unidimensional etiquetado capaz de contener cualquier tipo de dato, pueden pensar en ella como una

- columna en una hoja de cálculo o como un conjunto
- unidimensional de datos.



Pandas

Presentar Series y DataFrames como estructuras principales de Datos en Pandas

Etiquetas: Cada elemento en una Serie está asociado con una etiqueta o índice que puede ser numérico o basado en etiquetas.

Homogeneidad: todos los elementos en una Serie deben ser del mismo tipo de datos.

Creación de una Serie:

```
import pandas as pd

data = [1, 2, 3, 4, 5]
etiquetas = ['a', 'b', 'c', 'd', 'e']

serie = pd.Series(data, index=etiquetas)
```

Presentar Series y DataFrames como estructuras principales de Datos en Pandas

DataFrames:

Un DataFrame es una estructura bidimensional etiquetada con columnas que pueden contener diferentes tipos de datos, pueden pensar en él como una hoja de cálculo o una tabla SQL.

Columnas: cada columna en un DataFrame es esencialmente una Serie.

Índices: los DataFrames tienen índices tanto para filas como para columnas.

Creación de un DataFrame:

```
import pandas as pd

datos = {'Nombre': ['Alice', 'Bob', 'Charlie'],
        'Edad': [25, 30, 35],
        'Ciudad': ['A', 'B', 'C']}

df = pd.DataFrame(datos)
```



Ambas estructuras son esenciales en el análisis de datos y la manipulación en Pandas. Las Series son útiles para representar datos unidimensionales, mientras que los DataFrames permiten trabajar con datos bidimensionales, lo que facilita la manipulación y análisis de conjuntos de datos más complejos en el ámbito de la inteligencia artificial y la ciencia de datos.

Indexación Avanzada en Pandas: mostrar cómo utilizar etiquetas y posiciones para indexar en Pandas

La indexación avanzada en Pandas es una característica poderosa que permite acceder y manipular datos de manera más específica utilizando etiquetas y posiciones.

Indexación por Etiquetas:

Para acceder a datos utilizando etiquetas, pueden usar los métodos `loc[]` o `at[]`. El siguiente ejemplo ilustra cómo seleccionar una fila y una columna específicas por etiquetas:



```
import pandas as pd

datos = {'Nombre': ['Alice', 'Bob', 'Charlie'],
        'Edad': [25, 30, 35],
        'Ciudad': ['A', 'B', 'C']}

df = pd.DataFrame(datos, index=['a', 'b', 'c'])

# Seleccionar datos con loc[]
seleccion_loc = df.loc['b', 'Edad']
print(seleccion_loc) # Resultado: 30

# Seleccionar datos con at[]
seleccion_at = df.at['b', 'Edad']
print(seleccion_at) # Resultado: 30
```

Indexación Avanzada en Pandas: mostrar cómo utilizar etiquetas y posiciones para indexar en Pandas

Indexación por Posiciones:

Para acceder a datos utilizando posiciones, puedes usar los métodos `iloc[ ]` o `iat[ ]`. Aquí hay un ejemplo de cómo seleccionar datos basados en posiciones:



```
import pandas as pd

datos = {'Nombre': ['Alice', 'Bob', 'Charlie'],
        'Edad': [25, 30, 35],
        'Ciudad': ['A', 'B', 'C']}

df = pd.DataFrame(datos)

# Seleccionar datos con iloc[]
seleccion_iloc = df.iloc[1, 1]
print(seleccion_iloc) # Resultado: 30

# Seleccionar datos con iat[]
seleccion_iat = df.iat[1, 1]
print(seleccion_iat) # Resultado: 30
```



Estos métodos proporcionan flexibilidad para acceder a datos específicos en un DataFrame de Pandas, ya sea mediante etiquetas o posiciones numéricas. La elección entre ellos dependerá de tus necesidades específicas de indexación en el análisis de datos

Pandas

Operaciones Aritméticas en DataFrames de pandas

En Pandas, los DataFrames admiten una amplia variedad de operaciones aritméticas que facilitan el procesamiento y análisis de datos. Aquí hay un resumen de las operaciones aritméticas comunes en DataFrames:

Operaciones Elemento a Elemento:

Puedes realizar operaciones aritméticas elemento a elemento en un DataFrame. Por ejemplo, sumar un valor constante a todas las entradas del DataFrame:



```
import pandas as pd

datos = {'A': [1, 2, 3],
        'B': [4, 5, 6]}
df = pd.DataFrame(datos)

# Sumar 10 a todas las entradas
df_suma = df + 10
print(df_suma)
```

Pandas



Operaciones Aritméticas en DataFrames de pandas

Operaciones entre DataFrames:

También puedes realizar operaciones entre DataFrames, las operaciones se realizan en función de las etiquetas y las columnas coincidentes:



```
import pandas as pd

datos1 = {'A': [1, 2, 3],
          'B': [4, 5, 6]}
df1 = pd.DataFrame(datos1)

datos2 = {'A': [10, 20, 30],
          'B': [40, 50, 60]}
df2 = pd.DataFrame(datos2)

# Sumar DataFrames
df_suma = df1 + df2
print(df_suma)
```

Pandas



Operaciones Aritméticas en DataFrames de pandas

Operaciones con Series:

Las operaciones aritméticas también se pueden realizar entre DataFrames y Series; Pandas alinea automáticamente los índices antes de realizar la operación:



```
import pandas as pd

datos = {'A': [1, 2, 3],
        'B': [4, 5, 6]}
df = pd.DataFrame(datos)

serie = pd.Series([10, 20], index=['A', 'B'])

# Sumar DataFrame y Serie
resultado = df + serie
print(resultado)
```



Estas operaciones aritméticas son esenciales para realizar cálculos eficientes en conjuntos de datos de Pandas, facilitando el análisis y procesamiento de datos en el ámbito de la inteligencia artificial y la ciencia de datos.



TIC

TALENTO
TECH

AZ | PROYECTOS
EDUCATIVOS

