Map & Apply

MUKESH KUMAR

AGENDA

- What is df.map()?
- What is df.apply()?
- Example of df.apply() on Columns
- Example of df.apply() on Rows
- df.map() vs df.apply()
- When to Use Which?
- Practical Applications

What is df.map()?

- **Usage**: Can be used with both pandas Series and DataFrames.

- **Functionality**:
  - **For Series:** Applies a function element-wise to each element in the Series. It's ideal for simple transformations or substitutions.
  - **For DataFrames:** Applies a function element-wise to each element in the DataFrame.

What is df.map()?

- Syntax:

```
df.map(function/dictionary/series)
```

- **Input**: Accepts a dictionary, Series, or a callable function.

- **Return**: Returns a new Series (if applied to a Series) or a new DataFrame (if applied to a DataFrame) with the applied function.

Examples

- Refer Notebook:
  - Map_N_Apply.ipynb

What is df.apply()?

- **Usage**: Can be used with both Series and DataFrames.
- **Functionality:** Applies a function along an axis of the DataFrame (either rows or columns). It is more versatile and can handle complex operations.
- Syntax:

```
df.apply(function, axis=0/1)
```

  - **axis=0: Applies the function to columns.**
  - **axis=1: Applies the function to rows.**

What is df.apply()?

- **Input:** Accepts a function and optionally an axis (0 for rows, 1 for columns).
- **Return:** Returns a new DataFrame or Series depending on the axis and function applied.

Key Differences

- **Element-wise vs. Axis-wise**: map() is for element-wise transformations, while apply() operates along an axis (rows or columns).

- **Complexity**: apply() is more flexible and can handle complex functions that operate on entire rows or columns, whereas map() is better suited for simpler, element-wise transformations.

Practical Applications

- Refer Notebook
  - Practical_Applications.ipynb

Summary

- Use df.map() for element-wise transformations on Series or DataFrames.
- Use df.apply() when you need to apply a function along a specific axis (rows or columns) of a DataFrame or to a Series.