

Numpy & pandas

Contents

I. DataFrames	3
What is a Dataframe?	3
1. Creating a Pandas DataFrame	3
2. Viewing and Accessing Data in a DataFrame	3
3. Descriptive Statistics and Summary Information	3
4. Cleaning Data	4
5. Data Merging and Grouping	4
6. Working with Categorical Data	5
7. Read\Write a Dataframe	5
8. Visualization (Using Matplotlib)	5
II. Numpy	6
1. Array Creation	6
2. Array Indexing and Slicing	6
3. Array Manipulation and Reshaping	7
4. Mathematical and Statistical Operations	7
5. Copying and Broadcasting	7
6. Linear Algebra and Matrix Operations	8
7. Random Sampling	8
8. Set Operations	8

I. DataFrames

What is a Dataframe?

A dataframe is a 2-dimensional data structure that can store data of different types in columns, similar to a database or Excel spreadsheet. It consists of a dictionary of lists in which the list each have their own identifiers or keys, such as “last name” or “food group.”

DataFrames are most commonly used object in pandas

1. Creating a Pandas DataFrame

Method/Attribute	Description	Key Attributes
<code>pandas.DataFrame()</code>	Creates a DataFrame from provided data. Supports multiple input formats like lists, dictionaries, and arrays.	data: Input data (e.g., numpy array, list, dictionary, series). index: Custom row labels. columns: Custom column labels. dtype: Specifies data type for the DataFrame. If not provided, inferred automatically.

2. Viewing and Accessing Data in a DataFrame

Method/Attribute	Description
<code>df.head()</code>	Returns the first 5 rows (or specified number) of the DataFrame.
<code>df.tail()</code>	Returns the last 5 rows (or specified number) of the DataFrame.
<code>df["column_name"]</code>	Accesses a single column in the DataFrame.
<code>df.columns</code>	Returns the column labels of the DataFrame.
<code>df.dtypes</code>	Returns the data types of each column.
<code>df.info()</code>	Provides technical information about the DataFrame, including data types, non-null counts, and memory usage.

3. Descriptive Statistics and Summary Information

Method	Description	Key Attributes
<code>df.describe()</code>	Generates descriptive statistics for numerical columns (e.g., count, mean, standard deviation).	-

df.max()	Returns the maximum value for a specific column.	axis: If axis=0 (default), it calculates the max for each column. If axis=1, it calculates the max for each row.
----------	--	--

4. Cleaning Data

Method/Attribute	Description	Key Attributes
df.dropna()	Removes rows with missing values (NaN).	axis: axis=0 drops rows (default), axis=1 drops columns with missing values.
df.fillna()	Fills missing values with a specified value (or statistical measure).	-
df.duplicated()	Returns a boolean Series indicating whether each row is duplicated.	keep: keep='first' (default) keeps the first occurrence, keep='last' keeps the last, and keep=False marks all duplicates.
df.drop_duplicates()	Removes duplicate rows from the DataFrame.	subset: Defines which columns to consider for identifying duplicates.

5. Data Merging and Grouping

Method/Attribute	Description	Key Attributes
df.groupby()	Groups data by one or more columns and applies aggregation functions.	by: The column(s) to group by. Example: df.groupby("column_name").
pd.concat()	Concatenates multiple DataFrames row-wise or column-wise.	axis: axis=0 (default) for row-wise concatenation, axis=1 for column-wise.
pd.merge()	Merges two DataFrames using SQL-style joins based on common columns.	how: Defines join type (inner, outer, left, right). Default is inner. on: The common column(s) to merge on. Example: pd.merge(df1, df2, on='key').

6. Working with Categorical Data

Method/Attribute	Description	Key Attributes
<code>astype("category")</code>	Converts a column to a categorical data type.	-
<code>df["column"].cat</code>	Series-specific accessor for handling categorical data.	<code>rename_categories()</code> : Renames categories. <code>set_categories()</code> : Reorders or adds categories.
<code>df.sort_values()</code>	Sorts the DataFrame by values in one or more columns.	<code>by</code> : Specifies the column(s) to sort by. <code>ascending</code> : Boolean to determine sorting order.

7. Read\Write a Dataframe

Method/Attribute	Description	Key Attributes
<code>pd.read_csv()</code>	Loads a CSV file into a DataFrame.	
<code>df.to_excel()</code>	Exports the DataFrame to an Excel file.	<code>index</code> : Boolean that determines whether row names are written to the file. Default is True.

8. Visualization (Using Matplotlib)

Method/Attribute	Description	Key Attributes
<code>df.plot()</code>	Plots data in a DataFrame using matplotlib. Supports several plot types (line, bar, scatter, etc.).	<code>kind</code> : Specifies the type of plot (e.g., line, bar, scatter, hist, box).

II. Numpy

1. Array Creation

Method/ Attribute	Description	Key Attributes
numpy.array()	Creates an array from a list, tuple, or other array-like objects.	ndmin : Minimum number of dimensions for the array.
numpy.zeros()	Returns a new array of a specified shape filled with zeros.	-
numpy.ones()	Returns a new array of a specified shape filled with ones.	-
numpy.empty()	Returns a new array of a specified shape, but without initializing entries.	-
numpy.linspace()	Generates an array of evenly spaced values over a specified range.	start, stop : Defines the range of values. num : # of values to generate.
numpy.arange()	Returns an array of evenly spaced values within a given range.	start, stop, step : Defines the range and step size for values.
numpy.shape	Defines the dimensions of the array.	-
numpy.dtype	Specifies the data type.	-
numpy.full()	Creates a new array of a specified shape, filled with a given value.	-
numpy.eye()	Creates a 2D array with ones on the diagonal and zeros elsewhere (an identity matrix).	N : The number of rows and columns. k : The index of the diagonal.
numpy.random.rand()	Generates an array of random values from a uniform distribution over [0,1).	-
numpy.random.randn()	Generates random values from a standard normal distribution (mean 0, variance 1).	
numpy.meshgrid()	Generates coordinate matrices from coordinate vectors, often used for 2D/3D plotting or grid creation.	-

2. Array Indexing and Slicing

Method	Description	Key Attributes/Syntac
Array Indexing	Accesses elements of an array using zero-based indexing.	Syntax : array[index]. Negative indexing can be used to select elements from the end.
Array Slicing	Extracts a portion of an array using the start:stop:step notation.	Syntax : array[start:stop:step]. Default values: start = 0, stop = length, step = 1.

numpy.take()	Selects elements from an array along an axis using indices.	indices: Array of indices to select. axis: Axis along which to select elements.
---------------------	---	---

3. Array Manipulation and Reshaping

Method/Attribute	Description
numpy.reshape()	Reshapes an array without changing its data.
numpy.transpose()	Reverses or permutes the axes of an array.
numpy.ravel()	Flattens a multi-dimensional array into a one-dimensional array.
numpy.concatenate()	Joins a sequence of arrays along an existing axis.
numpy.split()	Splits an array into multiple sub-arrays.
numpy.hstack()	Stacks arrays in sequence horizontally (column-wise).
numpy.vstack()	Stacks arrays in sequence vertically (row-wise).
numpy.expand_dims()	Expands the shape of an array, inserting a new dimension.
numpy.squeeze()	Removes single-dimensional entries from the shape of an array.
numpy.tile()	Constructs an array by repeating the original array multiple times.

4. Mathematical and Statistical Operations

Method	Description
numpy.sum()	Computes the sum of array elements along a specified axis.
numpy.mean()	Computes the mean of array elements.
numpy.max()	Returns the maximum value of an array.
numpy.min()	Returns the minimum value of an array.
numpy.std()	Computes the standard deviation of array elements.
numpy.sqrt()	Returns the square root of each element in the array.
numpy.dot()	Computes the dot product of two arrays.
numpy.exp()	Computes the exponential of each element in the array.
numpy.log()	Computes the natural logarithm (base e) of each element in the array.
numpy.cumsum()	Computes the cumulative sum of the elements along a given axis.
numpy.cumprod()	Computes the cumulative product of the elements along a given axis.
numpy.diff()	Calculates the n-th order discrete difference along a given axis.
numpy.clip()	Limits the values in an array within a specified range (min and max). It takes a_min , a_max : Minimum and maximum values to clip the array values.

5. Copying and Broadcasting

Broadcasting is a powerful mechanism that allows NumPy to perform operations on arrays of different shapes. The smaller array is "broadcast" across the larger array so that they have compatible shapes.

Method	Description
--------	-------------

numpy.copy()	Returns a copy of the array.
numpy.add()	Adds two arrays element-wise, broadcasting if necessary.
numpy.subtract()	Subtracts the elements of one array from another, with broadcasting.
numpy.multiply()	Multiplies two arrays element-wise, broadcasting if necessary.
numpy.divide()	Divides the elements of one array by another, with broadcasting.
numpy.power()	Raises elements of an array to the power of the corresponding elements of another, with broadcasting.
numpy.mod()	Computes the element-wise remainder of division, with broadcasting.

6. Linear Algebra and Matrix Operations

Method/Attribute	Description
numpy.dot()	Computes the dot product of two arrays.
numpy.matmul()	Performs matrix multiplication between two arrays.
numpy.linalg.inv()	Computes the inverse of a square matrix.
numpy.linalg.det()	Computes the determinant of a matrix.
numpy.linalg.eig()	Computes the eigenvalues and eigenvectors of a square matrix.

7. Random Sampling

Method/Attribute	Description	Key Attributes
numpy.random.choice()	Generates a random sample from a given 1D array.	a : The input array. size : Output shape. replace : Boolean, whether sampling is with or without replacement.
numpy.random.randint()	Returns random integers from a specified range.	low, high : Defines the range of the random integers. size : Shape of the output array.

8. Set Operations

Method/Attribute	Description	Key Attributes
numpy.unique()	Returns the sorted unique elements of an array.	return_index : If True, returns the indices of the first occurrences of unique values.
numpy.intersect1d()	Finds the intersection of two arrays.	-
numpy.union1d()	Finds the union of two arrays.	-
numpy.setdiff1d()	Returns the set difference of two arrays.	-