

210
35

Neha Ganesht Karande
Batch No = Dg
Date = 15/11/24 Rajdhani

DATE / /

Class Assessment - 2

Explain.

Q. 1)

a) In Numpy, broadcasting is a powerful feature that allows arrays of different shapes & sizes to be combined or operated on together.

- The smaller array is "broadcast" across the larger array so that they have compatible shapes of element wise operators.

- The broadcasting rule in numpy is that dimensions are compatible when they are equal or one of them is 1.

- If dimensions of the arrays are not compatible Numpy will raise a 'value error'.

- eg:-

```
import numpy as np
```

```
arr1 = np.array([[1, 2, 3],  
                 [4, 5, 6]])
```

```
arr2 = np.array([10, 20, 30])
```

```
result = arr1 + arr2
```

```
print(result)
```

→ O/P :-

```
array([[11, 22, 33],  
       [14, 25, 36]])
```

Here each element of arr1 is added to the corresponding element of broadcasted 'arr2' leading to the final result.

Q. 2) In numpy both 'np.dot()' & 'np.matmul()' functions can be used for matrix multiplication but they have subtle differences in terms of their behaviour & usage.

a) 'np.dot()' :-

- The np.dot() funⁿ in numpy is a general purpose matrix multiplication funⁿ. It can perform dot products & matrix multiplications for 1-D & 2-D arrays.
- For 2-D arrays, it performs matrix multiplication & for 1-D arrays, it performs inner product (dot product).

b) 'np.matmul()' :-

- ⑧
- np.matmul() funⁿ is specifically designed for matrix multiplication.
 - It provides a clearer & more explicit syntax for matrix multiplication making the code more readable. In fact, 'np.matmul()' is equivalent to the @ operator in Python; introduced for matrix multiplication starting from python 3.5.

```
import pandas as pd
first_5_rows = sales_data.head()
print (first_5_rows)
```


b) `data-types = sales-data.dtypes`
`print (data-types)`

Q.4)

c)

`Import pandas as pd`

`sales_data ['Price-per-unit'] = 10.`

`sales_data ['Total-sales'] = sales_data ['Quantity-sold'] * [sales_data ['Price-per-unit']`

`print (sales_data)`

d) `sales_data ['Transaction-Date'] = pd.to_datetime`
`(sales_data ['Transaction-Date'])`
`print (Dales-data)`

Q.5)

`Import pandas as pd`

`average-quantity-per-product = sales_data.groupby`
`('product-ID')['Quantity-sold'].mean().`

Q.6)

a. Numerical python

Q.7)

b. `arr = numpy.array ([1,2,3])`

Q.8)

a. create an array filled with zeroes

Q.9)

a A 2-Dimensional (labelled) data structure.

Q.10)

c. `df ['column_name']`

Q.11)

b. `students-data ['Age']`

Q. 12) b. `Sum (sales_data ['price'] * sales_data ['quantity_sold'])`

Q. 13) a. A numpy is primarily used for data manipulation & mathematical operations on homogeneous arrays, while pandas provides high-level data structures & functions to manipulate & analyze structured data like DataFrames.

Q. 14) a. `df.iloc[:3]`

Q. 15) a. Drops all rows with missing values

Q. 16) a. `df.apply()`

Q. 17) a. `df.sort_values('column_name')`

Q. 18) b. Returns the largest n values in a specific column

Q. 19) c. `df.to_csv('output.csv')`

Q. 20) b. Converts a column to datetime format

Q. 21) a. `df.fillna()`