Batch :- 09

Name - mulla suhan Jameer
college - DACOE, karad

Page No. | Date

class Assessment :- 2

Annotate.

24/30

Q.1) Explain the concept of broadcasting in Numpy. provide an example?

→ Broadcasting in Numpy is a powerful mechanism that allows Numpy to perform operations on arrays of different shapes & sizes, without the need for explicit looping. The smaller array is "broadcast" across the larger array so that they have compatible shapes for element-wise operations.

1) If the arrays have a different no. of dimensions, pad the smaller array's shape with ones on its left side.
2) compare the sizes of the corresponding dimensions

e.g.
```
import numpy as np.
# creating a 3x3 array
arr-a = np.array([[1,2,3],
                  [4,5,6],
                  [7,8,9]])
# creating a 1x3 array
arr-b = np.array([10,20,30])
#Broadcasting arr-b to the shape of arr-a
result = arr-a + arr-b
# Displaying result
print (result)
```

In this example, 'arr-a' is a 3X3 array, and 'arr-b' is a 1X3 array, we can add these arrays directly, and Numpy will automatically broadcast 'arr-b' to the shape of 'arr-a' before performing the addition operation

The result will be:

```
[[11 , 22 , 33],
 [14 , 25 , 36],
 [17 , 28 ,39 ]]
```

Numpy automatically broadcast the smaller array 'arr-b' across the larger array 'arr-a' so that the shape becomes compatible for element-wise addition.

Q.2) Describe the difference betn np.dot() and np.matmul() in Numpy. When would you use each function?

→ Both 'np.dot()' & 'np.matmul()' functions in Numpy are used for matrix multiplication

1) 'NP. dot()' :-

i) 'np. dot (a,b)' computes the dot product of two arrays.

ii) For 2-D arrays, it performs matrix multiplication, & for 1-D arrays, it performs inner product (dot product).

e.g.
```
import numpy as np

A = np.array([[1,2], [3,4]])
B = np.array([[5,6], [7,8]])

result_dot = np.dot(A, B)
print (result_dot)
```

2) 'np.matmul()' :-

i) 'np.matmul(a,b)' also performs matrix multiplication, & it has the same behavior as 'np.dot():

e.g.
```
import numpy. as np

A = np.array([[1,2] [3,4]])
B = np.array([[5,6] [7,8]])

result_matmul = np.matmul(A, B)
print (result_matmul)
```

when to use each Function :-

i) use 'np.dot()' when working with 1-D arrays, as it computes the dot product for them.

ii) use 'np.matmul()' when you want to be explici about matrix multiplication & for better radability

```
Q.3)
→ a) import pandas as pd
     first_a_rows = sales_data.head()
     print (first_a_rows)

1/2  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~~
     sales_data ['Total_sales'] * 
             = sales_data ['quantity_sold'] *
               sales_data ['price_per_unit]

     print (sales_data)
1/2
  d) sales_data ['Transaction_Date'] = pd.to_dateme
                          (sales_data ['Transation_Date'])

     print (sales_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 zeros

Q.9) a) A two-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) Numpy is primarily used for data manipulation & mathematical operations on homogeneous arrays, while pandas provides high-level data structures & function 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

1 ✓

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

1 ✓

Q.20) b) converts a column to datatime format

1 ✓

Q.21) a) df.fillna()

1 ✓