

# INDIAN STATISTICAL INSTITUTE

Post Graduate Diploma in Business Analytics  
End Semester Examination, 2025

Computing for Data Sciences - Part I  
Maximum Marks: 51

Date of Examination: 19 December 2025, 2:30pm

Time Allowed: 1.5 Hours

Roll No:

## INSTRUCTIONS

- Consider the version of Python as Python 3 while answering questions on Python programming.

1. Answer any FIFTEEN questions.

[2 marks each  $\times$  15 = 30 marks]

- a) How do you check the data type of each column in a Pandas DataFrame named `df`?
- A. `df.data_types()`
  - B. `df.types`
  - C. `df.get_types`
  - D. `df.dtypes`
- b) Given `A = np.array([1, 2, 3])`, what is the shape of the array created by `np.newaxis` in the expression `A[:, np.newaxis]`?
- A. `ValueError`, because a 1D array cannot be indexed with two dimensions.
  - B. `(3,3)`
  - C. `(1,3)`
  - D. `(3,1)`
- c) What is the fundamental difference between a Pandas Series and a Pandas DataFrame?
- A. A Series can only hold integer data, while a DataFrame can hold multiple data types.
  - B. A Series is a 1-D labeled array, and a DataFrame is a 2-D labeled data structure.
  - C. A Series does not have an index, whereas a DataFrame does.
  - D. A Series can only store data from a CSV file, while a DataFrame can store data from any source.
- d) To calculate the mean of the 'Sales' column for each unique 'Region' in a DataFrame `df`, which of the following expressions is correct?
- A. `df.mean('Sales', by='Region')`
  - B. `df.groupby('Region')['Sales'].agg('mean')`
  - C. `df.aggregate('mean', groups='Region')`
  - D. `df.groupby('Region').mean()['Sales']`

- c) If `arr = np.array([1, 2, 3, 4, 5])` and `sub_arr = arr[1:4]`, and then execute `sub_arr[0] = 99`, what is the final value of `arr`?
- A. `[1, 99, 99, 99, 5]`
  - B. `[1, 2, 3, 4, 5]`
  - C. `[1, 99, 3, 4, 5]`
  - D. `[99, 2, 3, 4, 5]`
- f) You have a DataFrame `df` with a column 'Date' of object (string) type, formatted as 'YYYY-MM-DD'. What is the most memory-efficient way to convert this column to a datetime object?
- A. `df['Date'] = pd.to_datetime(df['Date'])`
  - B. `df['Date'] = df['Date'].apply(pd.to_datetime)`
  - C. `df['Date'] = df['Date'].astype('datetime64[ns]')`
  - D. `df['Date'] = pd.to_datetime(df['Date'], format='%Y-%m-%d', errors='coerce')`
- g) Given array `A` of shape `(4, 1)` and array `B` of shape `(5, )`, what is the shape of the result of the element-wise operation `A + B`, assuming broadcasting is successful?
- A. `(4, 1)`
  - B. `(4, 5)`
  - C. Broadcasting is not possible, a `ValueError` will occur.
  - D. `(5, 4)`
- h) `A = np.arange(10, 70, 10).reshape(2, 3)`. Which operation yields the element 50 using fancy indexing?
- A. `A[[1], [1]]`
  - B. `A[1, 1]`
  - C. `A[1][1]`
  - D. `A[[0, 1], [0, 1]]`
- i) Given `A = np.arange(6).reshape(2, 3)`, which NumPy operation correctly selects elements of `A` that are greater than 2 AND less than 5?
- A. `A[(A > 2) & (A < 5)]`
  - B. `A[np.where(A > 2, A < 5)]`
  - C. `A[(A > 2) | (A < 5)]`
  - D. `A[2 < A < 5]`
- j) Which of the following operations is a classic example of a Universal Function (ufunc) in NumPy designed for element-wise array operations?
- A. `np.dot(A, B)`
  - B. `np.sin(arr)`
  - C. `np.sum(arr, axis=0)`
  - D. `arr.sort()`
- k) What is the key difference in the output of `np.linspace(1, 10, 10)` versus `np.arange(1, 11)`?
- A. `np.arange` includes the endpoint 10 in its output, while `np.linspace` excludes it.
  - B. The output of `np.linspace(1, 10, 10)` will be integer, while `np.arange(1, 11)` will be float.
  - C. `np.linspace` includes the endpoint 10 in its output, while `np.arange` excludes its endpoint 11.
  - D. The two functions produce exactly the same 10 elements: `[1, 2, ..., 10]`.
- l) Given a 3D array `A` of shape `(D, H, W)`, representing (Depth, Height, Width) respectively, what is the resulting shape of the array after executing `A.transpose((2, 0, 1))`?
- A. `(H, W, D)`
  - B. `(D, W, H)`
  - C. `(W, D, H)`
  - D. `(D, H, W)`

- m) What is the default memory storage order for a newly created, multi-dimensional NumPy array, and why is this important for performance?
- Column-major (Fortran order), because it aligns better with linear algebra operations.
  - Randomized order, to prevent memory access patterns from being exploited.
  - Block-major, which is an optimized hybrid of row and column storage.
  - Row-major (C-order), which means iterating over the last axis is fastest due to memory contiguity.
- n) Given a  $4 \times 5$  array  $M$ , what does `np.sum(M, axis=0)` return, and what is its shape?
- An array of shape  $(5,)$  containing the sum of each column.
  - A scalar value which is the sum of all elements in the entire array.
  - A  $4 \times 5$  array where each element is the cumulative sum up to that point.
  - An array of shape  $(4,)$  containing the sum of each row.
- o) Given a DataFrame `df`, what does the code `df.iloc[0:2, 1]` return?
- The row at integer position 0 and the row at integer position 2, and the column at integer position 1.
  - The first two rows and the column at integer position 1.
  - Rows with index labels 0 and 1, and the column at integer position 1.
  - The first two rows and the column with label 1.
- p) What type of join is performed by default when `pd.merge(df1, df2, on="key_column")` is executed? `pd` is alias of pandas library; `df1`, and `df2` are DataFrames. The common column name of `df1` and `df2` is `key_column`.
- Right join
  - Outer join
  - Left join
  - Inner join
- q) Which one of the following is not a reduction operation (an operation that aggregates data and reduces the number of dimensions)?
- `np.argmax(A)`
  - `np.exp(A)`
  - `np.prod(A)`
  - `np.mean(A, axis=1)`
- r) What is the result of multiplying two arrays, `A = np.array([2, 4])` and `B = np.array([[1], [3]])`, using the element-wise multiplication operator `A * B`?
- A scalar value of 14 using the dot product
  - A  $2 \times 2$  array: `[[2, 6], [4, 12]]`
  - A `ValueError` due to incompatible shapes
  - A  $2 \times 2$  array: `[[2, 4], [6, 12]]`
- s) Using Matplotlib, after plotting a line with `ax.plot(x, y, label='Data')`, which final method must be called to ensure that the label text actually appears on the plot?
- `plt.draw_legend()`
  - `ax.show_labels()`
  - `ax.set_legend_visible(True)`
  - `ax.legend()`

17. If you want to visualize data points without connecting them with a line, often used to show the distribution of two variables, which function is the most appropriate?
- A. `plt.scatter(x, y)`
  - B. `plt.point(x, y)`
  - C. `plt.plot(x, y, linestyle='none')`
  - D. `plt.bar(x, y)`

2. Answer any THREE questions.

[7 marks each  $\times 3 = 21$  marks]

a) If the sum of two unit vectors is a unit vector, what is the difference of their magnitudes? Define *logit* function. State steps to use logit function for binary classification. [3 + 1 + 3 = 7]

- b) Derive the equation of a plane given its surface normal  $\hat{n} = (a, b, c)^T$ . A point  $(x, y)$  is transformed to  $(x', y')$  using following two equations.

$$x' = a_{11}x + a_{12}y + a_{13} \text{ and } y' = a_{21}x + a_{22}y + a_{23}.$$

Explain which of these coefficients  $a_{ij}$  are involved in translation, rotation, reflection and scaling of  $(x, y)$  to  $(x', y')$ . [3 + 4 = 7]

- c) The weights of all edges of a graph  $G$  are multiplied by  $\lambda_1$  to generate a graph  $G_1$ . Similarly, the weights of all edges of the same graph  $G$  are added by  $\lambda_2$  to generate a graph  $G_2$ . What are the relations between the shortest paths of  $\{G \text{ and } G_1\}$  and  $\{G \text{ and } G_2\}$ ? Comment on the characteristics of adjacency matrix of a directed graph, an undirected graph and a graph with loops. [2 + 2 + 3 = 7]

- d) Prove that a rectangle takes the shape of a square, if the area of the rectangle is fixed and while its perimeter is minimized. Write the expression for cross-entropy loss function given  $\pi$  and  $\hat{\pi}$  as true and predicted probability distribution, respectively, of a data set. [5 + 2 = 7]

**END OF PAPER - Part I**