الاسم / اسلام السيد رمزى الغرباوى

سيكشن / 1

CHAPTER 8

Multiple Choice Questions (MCQs)

1. Web routing and URL mapping

2. groupby()

3. Seaborn

4. Micro web framework

5. A model class

6. TensorFlow

7. Eigenvalues computation

8. It supports dynamic computation graphs.

True / False Questions

1. False
2. True
3. True
4. False
5. True
6. True

Short Answer / Conceptual Questions

1. Explain the difference between NumPy and SciPy

- **NumPy** provides basic tools for numerical computing, including arrays, vectorized operations, and basic linear algebra.

- **SciPy** builds on NumPy and provides advanced scientific functions such as optimization, integration, interpolation, statistics, and advanced linear algebra.

2. What is the purpose of the groupby() function in Pandas? Give an example

Answer:

groupby() is used to group rows based on a column and apply aggregations like sum, mean, or count.

Example:

```
df.groupby("Department")["Salary"].mean()
```

This computes the average salary per department.

3. Compare Flask and Django in terms of complexity and use cases.

Answer:

- **Flask** is lightweight, simple, and good for small applications and APIs.

- **Django** is a full-stack framework with built-in features (auth, admin, ORM), making it suitable for large, complex web applications

4. Why are tensors important in deep learning frameworks like PyTorch and TensorFlow?

Answer:

Tensors store multi-dimensional numerical data, run efficiently on GPUs, and support automatic

differentiation — all essential for training deep learning models.

5. **What is the difference between Matplotlib and Seaborn in visualization?**

Answer:

- **Matplotlib** is low-level and gives full control but requires more code.
- **Seaborn** is high-level, provides better styling, and includes statistical plots like heatmaps
  and pairplots.

#Problem 1

```
import numpy as np


arr = np.array([1,2,3,4,5,6,7,8,9,10])

mean = arr.mean()     # Use () to call the function
```

```
median = np.median(arr)    # Median function from numpy

std_dev = arr.std()     # Standard deviation

print(f"Mean: {mean}")

print(f"Median: {median}")

print(f"Standard Deviation: {std_dev}")
```

#Problem 2

```
import pandas as pd


students = pd.DataFrame({

"Name": ["Ali", "Mona", "Omar", "Sara"],

"Age": [20, 22, 19, 21],

"Score": [85, 92, 78, 88]

})

high_scores = students[students["Score"] > 80]
```

```
print("Students with score above 80:")
print(high_scores)
```

#Problem 3

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [1, 4, 9, 16, 25]

plt.plot(x, y, marker='o', linestyle='-', color='blue')
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("Line Graph of y = x^2")
plt.show()
```

#Problem 4

```
from flask import Flask
app = Flask(_name_)
@app.route("/hello")
def hello():
    return "Hello, Advanced Python!"
if _name_== "_main_":
app.run(debug=True)
```

#Problem 5

```
import torch


a = torch.tensor([1, 2, 3])
```

```
b = torch.tensor([4, 5, 6])

dot_product = torch.dot(a, b)
elementwise_mult = a * b
print(f"Dot Product: {dot_product}")
print(f"Element-wise Multiplication: {elementwise_mult}")
```