

Name: Mohan Krishna Balaji

Reg No:192324221

Course Code: CSA0406

Course Name: Fundamental Data Science

1. Scenario: You are working on a project that involves analyzing student performance data for a class of 32 students. The data is stored in a NumPy array named `student_scores`, where each row represents a student and each column represents a different subject. The subjects are arranged in the following order: Math, Science, English, and History. Your task is to calculate the average score for each subject and identify the subject with the highest average score.

Question: How would you use NumPy arrays to calculate the average score for each subject and determine the subject with the highest average score? Assume 4x4 matrix that stores marks of each student in given order.



```
main.py + 43gwexut5
1 import numpy as np
2
3 # Sample 4x4 matrix representing marks of 4 students in 4 subjects
4 marks = np.array([[85, 90, 78, 92],
5                  [88, 76, 95, 89],
6                  [90, 85, 80, 91],
7                  [70, 80, 75, 88]])
8
9 # Calculate the average score for each subject (columns)
10 average_scores = np.mean(marks, axis=0)
11
12 # Determine the subject with the highest average score
13 highest_average_index = np.argmax(average_scores)
14 highest_average_score = average_scores[highest_average_index]
15
16 print("Average Scores for Each Subject:", average_scores)
17 print("Subject with Highest Average Score:", highest_average_index + 1, "with a score of", highest_average_score)
18
```

STDIN

Input for the program (Optional)

Output:

Average Scores for Each Subject: [83.25 82.75 82. 90.]

Subject with Highest Average Score: 4 with a score of 90.0

2. Scenario: You are a data analyst working for a company that sells products online. You have been tasked with analyzing the sales data for the past month. The data is stored in a NumPy array.

Question: How would you find the average price of all the products sold in the past month? Assume 3x3 matrix with each row representing the sales for a different product

The screenshot shows the OneCompiler Python IDE interface. The editor on the left contains a Python script named `main.py` with the following code:

```
1 import numpy as np
2
3 # Sample sales data: rows represent products, columns represent sales over days
4 sales_data = np.array([[10, 20, 30],
5                        [15, 25, 35],
6                        [20, 30, 40]])
7
8 # Calculate the average price
9 total_sales = np.sum(sales_data)
10 number_of_sales = sales_data.size
11 average_price = total_sales / number_of_sales
12
13 print(f"The average price of products sold in the past month is: ${average_price:.2f}")
14
```

The right-hand pane shows the output of the program:

```
STDIN
Input for the program (Optional)

Output:
The average price of products sold in the past month is: $25.00
```

3. Scenario: You are working on a project that involves analyzing a dataset containing information about houses in a neighborhood. The dataset is stored in a CSV file, and you have imported it into a NumPy array named `house_data`. Each row of the array represents a house, and the columns contain various features such as the number of bedrooms, square footage, and sale price.

Question: Using NumPy arrays and operations, how would you find the average sale price of houses with more than four bedrooms in the neighborhood?

The screenshot shows the OneCompiler Python IDE interface. The editor on the left contains a Python script named `main.py` with the following code:

```
1 import numpy as np
2
3 # Sample data: number of bedrooms and corresponding sale prices
4 bedrooms = np.array([3, 5, 4, 6, 2, 7, 5])
5 sale_prices = np.array([250000, 350000, 300000, 450000, 200000, 500000, 400000])
6
7 # Filter sale prices for houses with more than 4 bedrooms
8 filtered_prices = sale_prices[bedrooms > 4]
9
10 # Calculate the average sale price
11 average_price = np.mean(filtered_prices)
12
13 print(f"The average sale price of houses with more than four bedrooms is: ${average_price:.2f}")
14
```

The right-hand pane shows the output of the program:

```
STDIN
Input for the program (Optional)

Output:
The average sale price of houses with more than four bedrooms is: $425000.00
```

4. Scenario: You are working on a project that involves analyzing the sales performance of a company over the past four quarters. The quarterly sales data is stored in a NumPy array named `sales_data`, where each element represents the sales amount for a specific quarter. Your task is to calculate the total sales for the year and determine the percentage increase in sales from the first quarter to the fourth quarter.

Question: Using NumPy arrays and arithmetic operations calculate the total sales for the year and determine the percentage increase in sales from the first quarter to the fourth quarter?

```
main.py + 43gwexut5
1 import numpy as np
2
3 # Sales data for each quarter
4 sales = np.array([15000, 20000, 25000, 30000]) # Sales for Q1, Q2, Q3, Q4
5
6 # Calculate total sales for the year
7 total_sales = np.sum(sales)
8
9 # Calculate percentage increase from Q1 to Q4
10 percentage_increase = ((sales[3] - sales[0]) / sales[0]) * 100
11
12 print(f"Total Sales for the Year: ${total_sales}")
13 print(f"Percentage Increase from Q1 to Q4: {percentage_increase:.2f}%")
14
```

STDIN

Input for the program (Optional)

Output:

Total Sales for the Year: \$90000
Percentage Increase from Q1 to Q4: 100.00%

5. **Scenario:** You are a data analyst working for a car manufacturing company. As part of your analysis, you have a dataset containing information about the fuel efficiency of different car models. The dataset is stored in a NumPy array named `fuel_efficiency`, where each element represents the fuel efficiency (in miles per gallon) of a specific car model. Your task is to calculate the average fuel efficiency and determine the percentage improvement in fuel efficiency between two car models.

Question: How would you use NumPy arrays and arithmetic operations to calculate the average fuel efficiency and determine the percentage improvement in fuel efficiency between two car models?

```
main.py + 43gwexut5
1 import numpy as np
2
3 # Fuel efficiencies in miles per gallon (mpg)
4 model_a = np.array([25, 27, 30]) # Model A efficiencies
5 model_b = np.array([30, 32, 35]) # Model B efficiencies
6
7 # Calculate average fuel efficiencies
8 avg_a = np.mean(model_a)
9 avg_b = np.mean(model_b)
10
11 # Calculate percentage improvement
12 percentage_improvement = ((avg_b - avg_a) / avg_a) * 100
13
14 print(f"Average Fuel Efficiency of Model A: {avg_a} mpg")
15 print(f"Average Fuel Efficiency of Model B: {avg_b} mpg")
16 print(f"Percentage Improvement: {percentage_improvement:.2f}%")
17
```

STDIN



Input for the program (Optional)



Output:

Average Fuel Efficiency of Model A: 27.333333333333332 mpg
Average Fuel Efficiency of Model B: 32.333333333333336 mpg
Percentage Improvement: 18.29%

6. **Scenario:** You are a cashier at a grocery store and need to calculate the total cost of a customer's purchase, including applicable discounts and taxes. You have the item prices and quantities in separate lists, and the discount and tax rates are given as percentages. Your task is to calculate the total cost for the customer.


Question: Use arithmetic operations to calculate the total cost of a customer's purchase, including discounts and taxes, given the item prices, quantities, discount rate, and tax rate?




[PRICING](#)[EDITOR](#)[CHALLENGES](#)[COMPANY](#)

main.py

+

43gwexut5 

 NEW PYTHON 

```
1 def calculate_total_cost(prices, quantities, discount_rate, tax_rate):
2     # Calculate subtotal
3     subtotal = sum(price * quantity for price, quantity in zip(prices, quantities))
4
5     # Apply discount
6     discount = subtotal * discount_rate
7     discounted_total = subtotal - discount
8
9     # Calculate tax
10    tax = discounted_total * tax_rate
11
12    # Final total
13    total_cost = discounted_total + tax
14    return total_cost
15
16 # Example usage
17 item_prices = [100, 200, 50] # Prices of items
18 item_quantities = [1, 2, 3] # Quantities of items
19 discount_rate = 0.1 # 10% discount
20 tax_rate = 0.05 # 5% tax
21
22 total = calculate_total_cost(item_prices, item_quantities, discount_rate, tax_rate)
23 print(f'Total cost of purchase: ${total:.2f}')
24
```

STDIN

Input for the program (Optional)

Output:

Total cost of purchase: \$614.25

```

main.py + 43gwexut5
1 import pandas as pd
2 from datetime import datetime, timedelta
3
4 # Sample sales data
5 data = {
6     'product_name': ['Product A', 'Product B', 'Product C', 'Product A', 'Product B', 'Product D'],
7     'sale_date': [
8         '2023-09-15', '2023-09-20', '2023-09-25',
9         '2023-10-01', '2023-10-05', '2023-10-10'
10    ]
11 }
12
13 # Create DataFrame
14 df = pd.DataFrame(data)
15 df['sale_date'] = pd.to_datetime(df['sale_date'])
16
17 # Define the date range for the past month
18 end_date = datetime.now()
19 start_date = end_date - timedelta(days=30)
20
21 # Filter data for the past month
22 filtered_sales = df[(df['sale_date'] >= start_date) & (df['sale_date'] <= end_date)]
23
24 # Count sales per product and get top 5
25 top_products = filtered_sales['product_name'].value_counts().head(5)
26
27 print(top_products)
28

```

STDIN

Input for the program (Optional)

Output:

Series([], Name: count, dtype: int64)

9. Scenario: You work for a real estate agency and have been given a dataset containing information about properties for sale. The dataset is stored in a Pandas DataFrame named `property_data`. The DataFrame has columns for property ID, location, number of bedrooms, area in square feet, and listing price. Your task is to analyze the data and answer specific questions about the properties.

Question: Using Pandas DataFrame operations, how would you find the following information from the `property_data` DataFrame:

1. The average listing price of properties in each location.
2. The number of properties with more than four bedrooms.
3. The property with the largest area

```

main.py + 43gwexut5
1 import pandas as pd
2
3 # Sample DataFrame creation (replace this with your actual DataFrame)
4 property_data = pd.DataFrame({
5     'location': ['Location A', 'Location B', 'Location A', 'Location C'],
6     'listing_price': [300000, 450000, 350000, 500000],
7     'bedrooms': [3, 5, 4, 6],
8     'area': [1500, 2000, 1800, 2500]
9 })
10
11 # 1. Average listing price of properties in each location
12 average_price = property_data.groupby('location')['listing_price'].mean()
13
14 # 2. Number of properties with more than four bedrooms
15 properties_with_more_than_four_bedrooms = property_data[property_data['bedrooms'] > 4].shape[0]
16
17 # 3. Property with the largest area
18 largest_property = property_data.loc[property_data['area'].idxmax()]
19
20 # Display results
21 print("Average Listing Price by Location:\n", average_price)
22 print("Number of Properties with More than Four Bedrooms:", properties_with_more_than_four_bedrooms)
23 print("Property with the Largest Area:\n", largest_property)
24

```

STDIN

Input for the program (Optional)

Output:

Average Listing Price by Location:

location	listing_price
Location A	325000.0
Location B	450000.0
Location C	500000.0

Name: listing_price, dtype: float64

Number of Properties with More than Four Bedrooms: 2

Property with the Largest Area:

location	listing_price	bedrooms	area
Location C	500000	6	2500

Name: 3, dtype: object

10. Scenario: You are working on a data visualization project and need to create basic plots using Matplotlib. You have a dataset containing the monthly sales data for a company, including the month and corresponding sales values. Your task is to develop a Python program that generates line plots and bar plots to visualize the sales data.

Question:

1. How would you develop a Python program to create a line plot of the monthly sales data?
2. How would you develop a Python program to create a bar plot of the monthly sales data?

```

import matplotlib.pyplot as plt

# Monthly sales dataset
months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
sales = [2500, 2700, 3000, 2800, 3500, 4000, 4200, 3900, 3700, 3600, 3800, 4100]

# -----
# 1. Line Plot of Sales Data
# -----
plt.figure(figsize=(10, 5))
plt.plot(months, sales, marker='o', color='green', linestyle='-', linewidth=2)
plt.title('Monthly Sales - Line Plot')
plt.xlabel('Month')
plt.ylabel('Sales')
plt.grid(True)
plt.tight_layout()
plt.show()

# -----
# 2. Bar Plot of Sales Data
# -----
plt.figure(figsize=(10, 5))
plt.bar(months, sales, color='orange', edgecolor='black')
plt.title('Monthly Sales - Bar Plot')
plt.xlabel('Month')
plt.ylabel('Sales')
plt.grid(axis='y', linestyle='--')
plt.tight_layout()
plt.show()

```

Click Run or press **shift + ENTER** to run code

Enable code completions

