DSA0410 – Fundamentals of Data Science

Day - 1

Name : Mithlesh N

Reg.no.: 192424364

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.

```python
#exp1
import numpy as np

student_scores = np.loadtxt(
    "student_scores.csv",
    delimiter=",",
    skiprows=1,
    usecols=(1, 2, 3, 4)
)

subject_averages = np.mean(student_scores, axis=0)

subjects = ["Math", "Science", "English", "History"]

highest_avg_index = np.argmax(subject_averages)
highest_avg_subject = subjects[highest_avg_index]

for subject, avg in zip(subjects, subject_averages):
    print(f"{subject} Average Score: {avg:.2f}")

print(f"\nSubject with highest average score: {highest_avg_subject}")
```

```
Math Average Score: 81.97
Science Average Score: 82.56
English Average Score: 82.97
History Average Score: 82.88

Subject with highest average score: English
```

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

```
#exp2
import pandas as pd
import numpy as np

df = pd.read_csv("Sales.csv")

# Select only numeric columns automatically
sales_data = df.select_dtypes(include=[np.number]).to_numpy()

average_price = np.mean(sales_data)
print("Average price of all products sold:", average_price)
```

```
Average price of all products sold: 1883.1075483333334
```

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?

```
#exp3
import numpy as np

house_data = np.loadtxt(
    "House_Prediction.csv",
    delimiter=",",
    skiprows=1
)
houses_more_than_4 = house_data[house_data[:, 0] > 4]
average_sale_price = np.mean(houses_more_than_4[:, 2])

print("Average sale price of houses with more than 4 bedrooms:", average_sale_price)
```

```
Average sale price of houses with more than 4 bedrooms: 2.1591248135256094
```

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?

```python
#exp4
import pandas as pd
import numpy as np

# Load CSV
df = pd.read_csv("Sales.csv")

# Extract only numeric quarterly sales columns
sales_data = df.select_dtypes(include=[np.number]).to_numpy().flatten()
# Total sales for the year
total_sales = np.sum(sales_data)

# Percentage increase from first quarter to fourth quarter
percentage_increase = ((sales_data[-1] - sales_data[0]) / sales_data[0]) * 100

print("Total sales for the year:", total_sales)
print("Percentage increase from Q1 to Q4:", percentage_increase, "%")
```

```
Total sales for the year: 11298645.29
Percentage increase from Q1 to Q4: -99.98669201520912 %
```

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?

```python
#exp5
import pandas as pd
import numpy as np

# Load the dataset
df = pd.read_csv("Fuel_Efficiency datasets.csv")

# Select only numeric columns (MPG values)
fuel_efficiency = df.select_dtypes(include=[np.number]).to_numpy().flatten()
average_efficiency = np.mean(fuel_efficiency)
print("Average fuel efficiency:", average_efficiency, "MPG")
percentage_improvement = (
    (fuel_efficiency[-1] - fuel_efficiency[0]) / fuel_efficiency[0]
) * 100

print("Percentage improvement in fuel efficiency:", percentage_improvement, "%")
```

```
Average fuel efficiency: nan MPG
Percentage improvement in fuel efficiency: 7995.999999999999 %
```