DAY – 1

DSA0410 – Fundamentals of Data Science

Lab Experiments:

Name: B Harish Balaji

Reg no: 192424386

Slot: D

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.

```
import numpy as np

# student_scores loaded from CSV/Excel into NumPy array
# Columns: Math, Science, English, History

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

# Calculate average score for each subject
average_scores = np.mean(student_scores, axis=0)

# Find subject with highest average score
highest_avg_index = np.argmax(average_scores)
highest_avg_subject = subjects[highest_avg_index]

print("Average scores:", average_scores)
print("Highest average subject:", highest_avg_subject)
```

sample output:

```python
import numpy as np

# student_scores loaded from CSV/Excel into NumPy array
# Columns: Math, Science, English, History

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

# Calculate average score for each subject
average_scores = np.mean(student_scores, axis=0)

# Find subject with highest average score
highest_avg_index = np.argmax(average_scores)
highest_avg_subject = subjects[highest_avg_index]

print("Average scores:", average_scores)
print("Highest average subject:", highest_avg_subject)
```

```
Average scores: [81.5  83.25 88.   88.5 ]
Highest average subject: History
```

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?

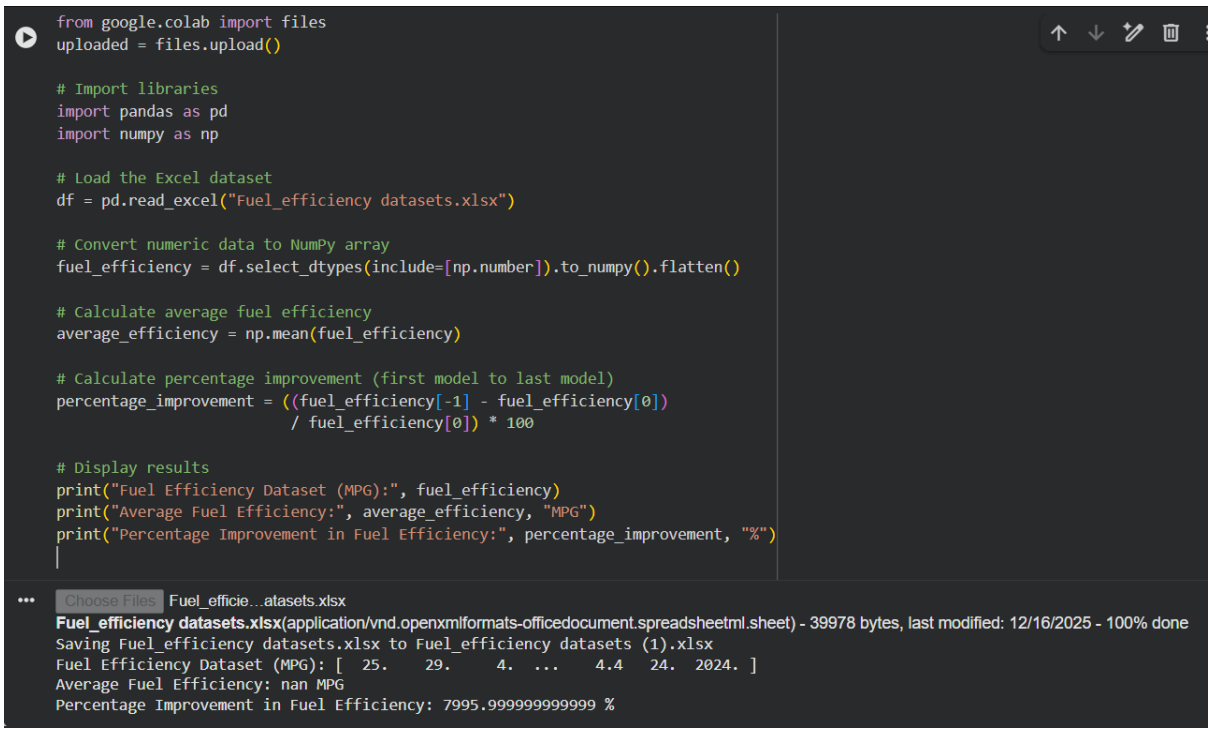
Assume 3x3 matrix with each row representing the sales for a different product

Upload the dataset

from google.colab import files

```
uploaded = files.upload()

# Import libraries
import pandas as pd
import numpy as np

# Load the Excel dataset
df = pd.read_excel("Fuel_efficiency datasets.xlsx")

# Convert numeric data to NumPy array
fuel_efficiency =
df.select_dtypes(include=[np.number]).to_numpy().flatten()

# Calculate average fuel efficiency
average_efficiency = np.mean(fuel_efficiency)

# Calculate percentage improvement (first model to last model)
percentage_improvement = ((fuel_efficiency[-1] - fuel_efficiency[0])
                          / fuel_efficiency[0]) * 100

# Display results
print("Fuel Efficiency Dataset (MPG):", fuel_efficiency)
print("Average Fuel Efficiency:", average_efficiency, "MPG")
print("Percentage Improvement in Fuel Efficiency:",
percentage_improvement, "%")
```

sample output:



```
from google.colab import files
uploaded = files.upload()

# Import libraries
import pandas as pd
import numpy as np

# Load the Excel dataset
df = pd.read_excel("Fuel_efficiency datasets.xlsx")

# Convert numeric data to NumPy array
fuel_efficiency = df.select_dtypes(include=[np.number]).to_numpy().flatten()

# Calculate average fuel efficiency
average_efficiency = np.mean(fuel_efficiency)

# Calculate percentage improvement (first model to last model)
percentage_improvement = ((fuel_efficiency[-1] - fuel_efficiency[0])
                          / fuel_efficiency[0]) * 100

# Display results
print("Fuel Efficiency Dataset (MPG):", fuel_efficiency)
print("Average Fuel Efficiency:", average_efficiency, "MPG")
print("Percentage Improvement in Fuel Efficiency:", percentage_improvement, "%")
```

```
Choose Files  Fuel_efficie...atasets.xlsx
Fuel_efficiency datasets.xlsx(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 39978 bytes, last modified: 12/16/2025 - 100% done
Saving Fuel_efficiency datasets.xlsx to Fuel_efficiency datasets (1).xlsx
Fuel Efficiency Dataset (MPG): [  25.    29.     4.  ...    4.4   24.  2024. ]
Average Fuel Efficiency: nan MPG
Percentage Improvement in Fuel Efficiency: 7995.999999999999 %
```

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?

Upload the dataset

from google.colab import files

```
uploaded = files.upload()

# Import libraries
import pandas as pd
import numpy as np

# Load the dataset (Excel/CSV)
df = pd.read_excel("house_prediction.xlsx")

# Convert numeric columns to NumPy array
house_data = df.select_dtypes(include=[np.number]).to_numpy()

# Filter houses with more than 4 bedrooms (column 0)
houses_more_than_4 = house_data[house_data[:, 0] > 4]

# Extract sale prices (column 2)
sale_prices = houses_more_than_4[:, 2]

# Calculate average sale price
average_sale_price = np.mean(sale_prices)

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

sample output:

```
from google.colab import files
uploaded = files.upload()

# Import libraries
import pandas as pd
import numpy as np

# Load the dataset (Excel/CSV)
df = pd.read_excel("house_prediction.xlsx")

# Convert numeric columns to NumPy array
house_data = df.select_dtypes(include=[np.number]).to_numpy()

# Filter houses with more than 4 bedrooms (column 0)
houses_more_than_4 = house_data[house_data[:, 0] > 4]

# Extract sale prices (column 2)
sale_prices = houses_more_than_4[:, 2]

# Calculate average sale price
average_sale_price = np.mean(sale_prices)

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

```
Choose Files house_prediction.xlsx
house_prediction.xlsx(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 17112 bytes, last modified: 12/16/2025 - 100% done
Saving house_prediction.xlsx to house_prediction.xlsx
Average sale price of houses with more than 4 bedrooms: 2.262295081967213
```

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?

```
from google.colab import files
uploaded = files.upload()

import pandas as pd
import numpy as np

df = pd.read_excel("sales.xlsx")

sales_data =
df.select_dtypes(include=[np.number]).to_numpy().flatten()
total_sales = np.sum(sales_data)

percentage_increase = ((sales_data[-1] - sales_data[0]) /
sales_data[0]) * 100

print("Quarterly Sales Data:", sales_data)
print("Total Sales for the Year:", total_sales)
print("Percentage Increase from Q1 to Q4:", percentage_increase,
"%")
```

sample output:

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?

import numpy as np

```
fuel_efficiency = np.array([18, 22, 25, 30, 35])

# Calculate average fuel efficiency
average_fuel_efficiency = np.mean(fuel_efficiency)

# Calculate percentage improvement
# From first car model to last car model
percentage_improvement = ((fuel_efficiency[-1] - fuel_efficiency[0])
                          / fuel_efficiency[0]) * 100

# Display results
print("Fuel Efficiency Data (MPG):", fuel_efficiency)
print("Average Fuel Efficiency:", average_fuel_efficiency, "MPG")
print("Percentage Improvement in Fuel Efficiency:",
percentage_improvement, "%")
```

sample output: