

# LAB EXPERIMENTATIONS

## DSA 0412 – FUNDAMENTALS OF DATA SCIENCE

Sk Abdul Azeez  
192472015

EXP 1 :

PROGRAM :

```
import numpy as np

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

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

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

highest_subject = subjects[np.argmax(average_scores)]

print("Average scores:", np.round(average_scores,2))

print("Subject with highest average score:", highest_subject)
```

OUTPUT :

```
exp1.ipynb
```

```
File Edit View Insert Runtime Tools Help
```

```
Commands + Code + Text Run all
```

```
[ ] Start coding or generate with AI.
```

```
import numpy as np

student_scores = np.loadtxt("student_scores.csv", delimiter=",", skiprows=1, usecols=(1, 2, 3, 4))
subjects = ["Math", "Science", "English", "History"]

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

highest_subject = subjects[np.argmax(average_scores)]

print("Average scores:", np.round(average_scores,2))
print("Subject with highest average score:", highest_subject)
```

```
... Average scores: [81.97 82.56 82.97 82.88]
Subject with highest average score: English
```

Variables Terminal Initialise Python 3

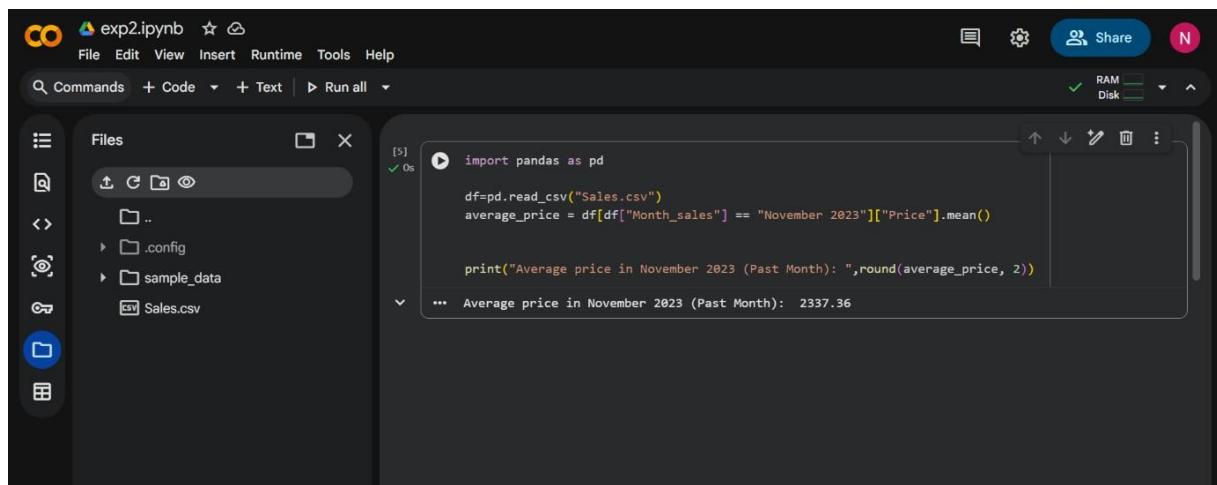
## EXP 2 :

### PROGRAM:

```
import pandas as pd  
  
df=pd.read_csv("Sales_data.csv")
```

```
average_price = df[df["Month_sales"]==" November 2023"]["Price"].mean()  
  
print("Average price in November 2023 (Past Month): ",round(average_price, 2))
```

### OUTPUT:



The screenshot shows a Jupyter Notebook interface with a dark theme. The left sidebar displays a file tree with 'exp2.ipynb' as the active notebook, containing files like '.config', 'sample\_data', and 'Sales.csv'. The main area shows the Python code and its execution output. The code imports pandas, reads a CSV file, filters for November 2023, calculates the average price, and prints the result. The output cell shows the command and the resulting average price of 2337.36.

```
import pandas as pd  
  
df=pd.read_csv("Sales.csv")  
average_price = df[df["Month_sales"] == " November 2023"]["Price"].mean()  
  
print("Average price in November 2023 (Past Month): ",round(average_price, 2))
```

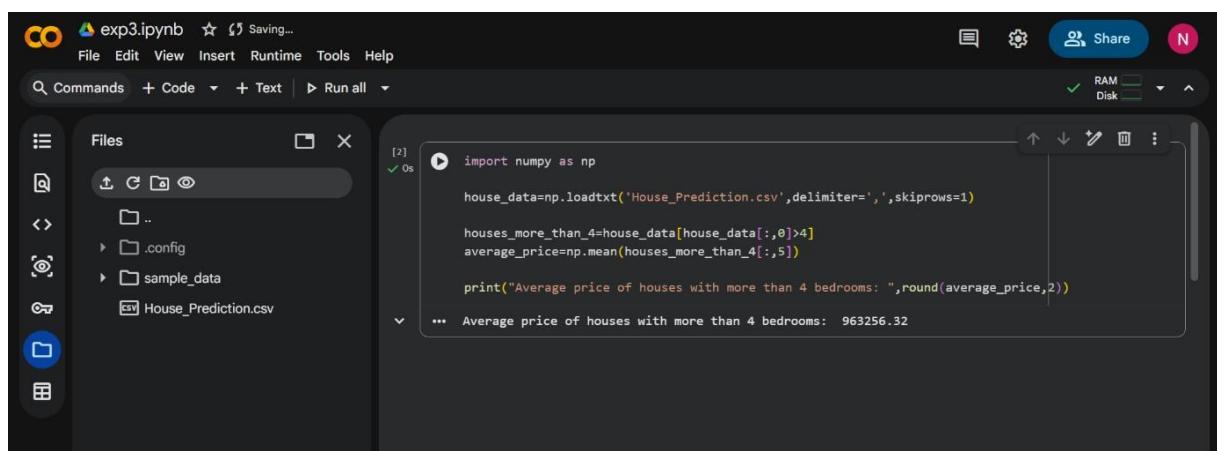
... Average price in November 2023 (Past Month): 2337.36

## EXP 3:

### PROGRAM :

```
import numpy as np  
  
house_data=np.loadtxt('House_data.csv',delimiter=',',skiprows=1)  
  
houses_more_than_4=house_data[house_data[:,0]>4]  
  
average_price=np.mean(houses_more_than_4[:,5])  
  
print("Average price of houses with more than 4 bedrooms: ",round(average_price,2))
```

### OUTPUT:



The screenshot shows a Jupyter Notebook interface with a dark theme. The left sidebar displays a file tree with 'exp3.ipynb' as the active notebook, containing files like '.config', 'sample\_data', and 'House\_Prediction.csv'. The main area shows the Python code and its execution output. The code loads a CSV file, filters for houses with more than 4 bedrooms, calculates the average price, and prints the result. The output cell shows the command and the resulting average price of 963256.32.

```
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_price=np.mean(houses_more_than_4[:,5])  
  
print("Average price of houses with more than 4 bedrooms: ",round(average_price,2))
```

... Average price of houses with more than 4 bedrooms: 963256.32

## EXP 4:

### PROGRAM :

```
import numpy as np

sales_data = np.genfromtxt("Sales_data.csv", delimiter=",", skip_header=1, dtype=str)

months = sales_data[:, 1]

sales = sales_data[:, 4].astype(float)

Q1 = 0.0

Q4 = 0.0

for i in range(len(months)):

    if ("January" in months[i]) or ("February" in months[i]) or ("March" in months[i]):

        Q1 += sales[i]

    elif ("October" in months[i]) or ("November" in months[i]) or ("December" in months[i]):

        Q4 += sales[i]

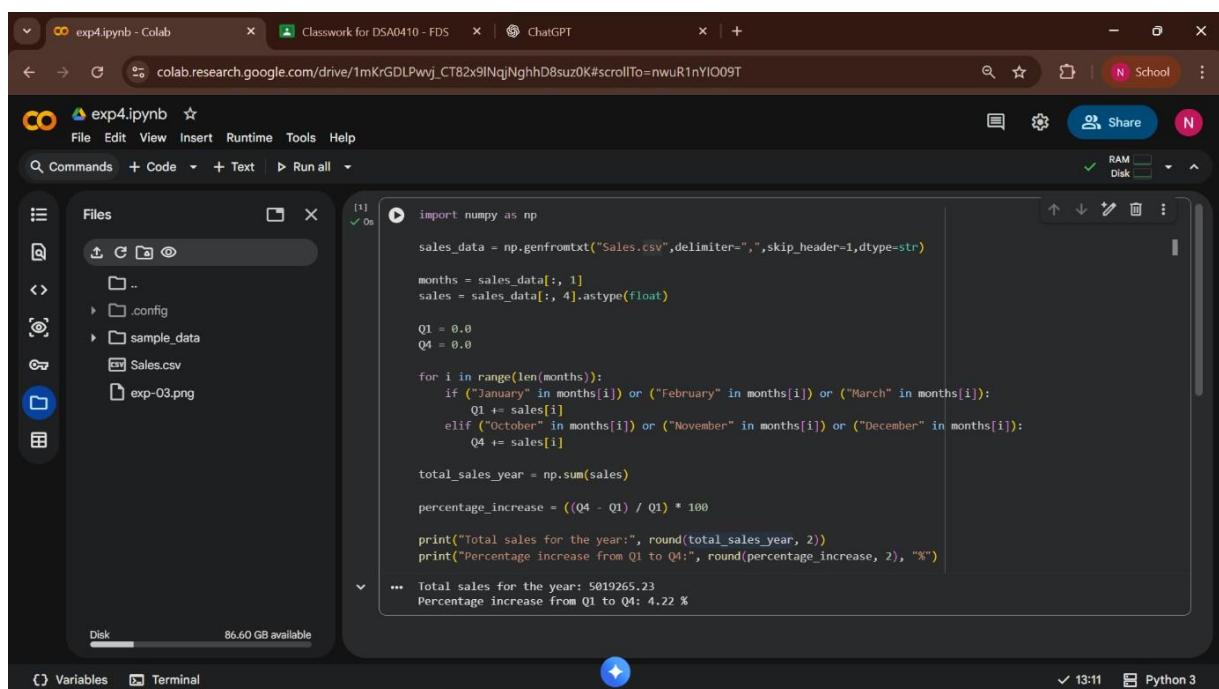
total_sales_year = np.sum(sales)

percentage_increase = ((Q4 - Q1) / Q1) * 100

print("Total sales for the year:", round(total_sales_year, 2))

print("Percentage increase from Q1 to Q4:", round(percentage_increase, 2), "%")
```

### OUTPUT:



```
File Edit View Insert Runtime Tools Help
Q Commands + Code + Text Run all
Files
Disk 86.60 GB available
[1] exp4.ipynb
import numpy as np

sales_data = np.genfromtxt("Sales.csv", delimiter=",", skip_header=1, dtype=str)

months = sales_data[:, 1]
sales = sales_data[:, 4].astype(float)

Q1 = 0.0
Q4 = 0.0

for i in range(len(months)):
    if ("January" in months[i]) or ("February" in months[i]) or ("March" in months[i]):
        Q1 += sales[i]
    elif ("October" in months[i]) or ("November" in months[i]) or ("December" in months[i]):
        Q4 += sales[i]

total_sales_year = np.sum(sales)

percentage_increase = ((Q4 - Q1) / Q1) * 100

print("Total sales for the year:", round(total_sales_year, 2))
print("Percentage increase from Q1 to Q4:", round(percentage_increase, 2), "%")
... Total sales for the year: 5019265.23
Percentage increase from Q1 to Q4: 4.22 %
```

## EXP 5:

### PROGRAM :

```
import numpy as np

fuel_data=np.genfromtxt('Fuel_data.csv', delimiter=',', skip_header=1,dtype=str)

fuel_efficiency = fuel_data[:, 7].astype(float)

make = fuel_data[:,8]

average_efficiency = np.mean(fuel_efficiency)

mazda_eff = fuel_efficiency[make == "mazda"]

audi_eff = fuel_efficiency[make == "audi"]

mazda_avg = np.mean(mazda_eff)

audi_avg = np.mean(audi_eff)

percentage_improvement = ((mazda_avg - audi_avg) / audi_avg) * 100

print("Average Fuel Efficiency of all cars:", round(average_efficiency, 2), "MPG")

print("Average Fuel Efficiency of Mazda:", round(mazda_avg, 2), "MPG")

print("Average Fuel Efficiency of Audi:", round(audi_avg, 2), "MPG")

print("Percentage Improvement from Mazda to Audi:", round(percentage_improvement, 2), "%")
```

### OUTPUT:

The screenshot shows a Google Colab interface with the following details:

- Title Bar:** exp5.ipynb - Colab, ChatGPT
- File Explorer:** Shows a directory structure with files like .config, sample\_data, and Fuel\_Efficiency.csv.
- Code Editor:** Displays the Python code provided in the question.
- Output Area:** Shows the execution results:

```
... Average Fuel Efficiency of all cars: 28.61 MPG
Average Fuel Efficiency of Mazda: 33.48 MPG
Average Fuel Efficiency of Audi: 27.97 MPG
Percentage Improvement from Mazda to Audi: 19.7 %
```
- Bottom Status Bar:** Variables, Terminal, 13:16, Python 3

