## GenAi CA 2

Name: Om Bonde

PRN: 21070521047

Section A

Q:5 Generate a model for Covid 19 with symptoms of parameters like fever, cold, shivering,

weight loss, generate 100 model data with random values for each parameter and order by

parameter lowest to highest in display based on the input parameter.

import random

import pandas as pd

def generate\_covid\_data(num\_cases):

return [

{

"fever": random.randint(95, 105), # fever: Temperature in Fahrenheit. Values range from 95 to 105.

"cold": random.randint(0, 10), # cold: Severity of cold symptoms.

# A value of 0 might represent no cold symptoms, while 10 could indicate severe cold.

"shivering": random.randint(0, 5), # shivering: Frequency or intensity of shivering.

# 0 might mean no shivering, while 10 could represent frequent or intense shivering.

"weight\_loss": random.randint(0, 10) # weight\_loss: Amount of weight loss in kg.

}

for \_ in range(num\_cases)

]

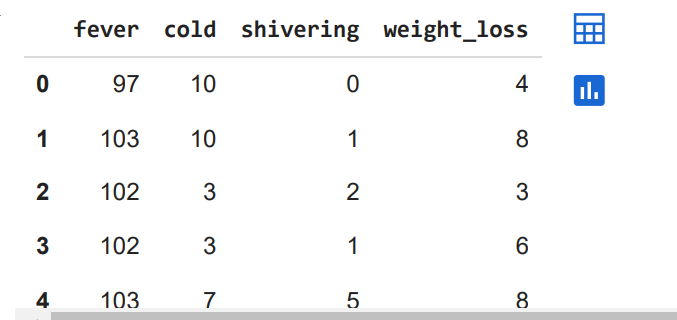
def sort\_data(data, parameter):

return sorted(data, key=lambda x: x[parameter])

data = generate\_covid\_data(100)

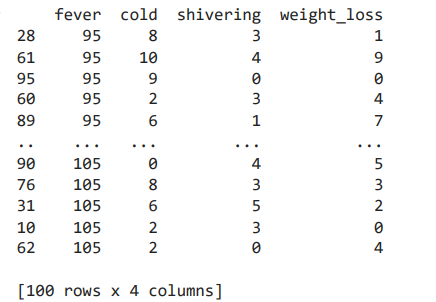
df = pd.DataFrame(data)

df.head()



sorted\_df = df.sort\_values(by="fever")

print(sorted\_df)



sorted\_df = df.sort\_values(by="weight\_loss")

print(sorted\_df)

A screenshot of a computer code

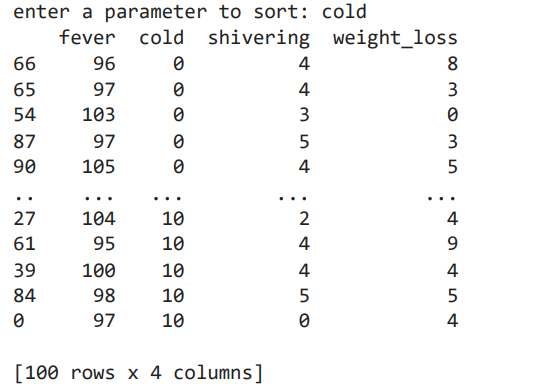
Description automatically generated

a = input("enter a parameter to sort: ")

sorted\_df = df.sort\_values(by= a )

print(sorted\_df)

Output:



Q:6 Generate a model to represent a mathematical equation, write a program to parse the

equation, and ask for input for each parameter.

import numpy as np

def parse\_equation(equation):

return equation

def get\_input\_for\_parameters(equation):

params = [param for param in equation if param.isalpha()] # Assuming parameters are single letters

values = {param: float(input(f"Enter the value for {param}: ")) for param in params}

return values

def evaluate\_equation(equation, values):

for param, value in values.items():

equation = equation.replace(param, str(value))

return eval(equation)

if \_\_name\_\_ == "\_\_main\_\_":

equation = input("Enter the mathematical equation: ")

values = get\_input\_for\_parameters(equation)

result = evaluate\_equation(equation, values)

print("Result:", result)

Output:

