AI Assisted Coding Lab 2.1

M Abhiram
2303A51667
Batch-18

Task 1: Statistical Summary for Survey Data

❖ Scenario:

You are a data analyst intern working with survey responses stored as numerical lists.

❖ Task:

Use Google Gemini in Colab to generate a Python function that reads a list of numbers and calculates the mean, minimum, and maximum values.

❖ Expected Output:

➢ Correct Python function

➢ Output shown in Colab

➢ Screenshot of Gemini prompt and result

```
# Use Google Gemini in Colab to generate a Python function that reads a
list
# of numbers and calculates the mean, minimum, and maximum values.
def calculate_stats(numbers):
    if not numbers:
        return None, None, None
    mean = sum(numbers) / len(numbers)
    minimum = min(numbers)
    maximum = max(numbers)
    return mean, minimum, maximum
# Example usage
numbers = [10, 23, 45, 66, 78, 89, 90, 12, 34, 57]
mean, minimum, maximum = calculate_stats(numbers)
print(f"Mean: {mean}, Min: {minimum}, Max: {maximum}")
```

Output:

Mean: 50.4, Min: 10, Max: 90

Task 2: Armstrong Number – AI Comparison

❖ Scenario:

You are evaluating AI tools for numeric validation logic.

❖ Task:

Generate an Armstrong number checker using Gemini and GitHub Copilot.

Compare their outputs, logic style, and clarity.

❖ Expected Output:

➢ Side-by-side comparison table

➢ Screenshots of prompts and generated code

```
# Generate an Armstrong number checker using Gemini and GitHub
# Copilot.
# Compare their outputs, logic style, and clarity.
def is_armstrong(number):
    num_str = str(number)
    num_digits = len(num_str)
    sum_of_powers = sum(int(digit) ** num_digits for digit in num_str)
    return sum_of_powers == number
number = int(input("Enter a number to check if it's an Armstrong number:
"))
if is_armstrong(number):
    print(f"{number} is an Armstrong number.")
else:
    print(f"{number} is not an Armstrong number.")
```

Output:

Enter a number to check if it's an Armstrong number: 20

20 is not an Armstrong number.

Task 3: Leap Year Validation Using Cursor AI

❖ Scenario:

You are validating a calendar module for a backend system.

❖ Task:

Use Cursor AI to generate a Python program that checks whether a given year is a leap year.

Use at least two different prompts and observe changes in code.

❖ Expected Output:

➢ Two versions of code

➢ Sample inputs/outputs

➢ Brief comparison

```
# Use Cursor AI to generate a Python program that checks whether a given
# year is a leap year.
# Use at least two different prompts and observe changes in code.
def is_leap_year(year):
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
        return True
    else:
        return False
year = int(input("Enter a year to check if it's a leap year: "))
if is_leap_year(year):
    print(f"{year} is a leap year.")
else:
    print(f"{year} is not a leap year.")
```

Output:

Enter a year to check if it's a leap year: 2016

2016 is a leap year.

Task 4: Student Logic + AI Refactoring (Odd/Even Sum)

❖ Scenario:

Company policy requires developers to write logic before using AI.

❖ Task:

Write a Python program that calculates the sum of odd and even numbers in a tuple, then refactor it using any AI tool.

❖ Expected Output:

➢ Original code

➢ Refactored code

➢ Explanation of improvements

```
# Write a Python program that calculates the sum of odd and even numbers
# in a tuple, then refactor it using any AI tool.
numbers_tuple = (10, 23, 45, 66, 78, 89, 90, 12, 34, 57)
even_sum = 0
odd_sum = 0
for number in numbers_tuple:
    if number % 2 == 0:
        even_sum += number
    else:
        odd_sum += number
print(f"Sum of even numbers: {even_sum}")
print(f"Sum of odd numbers: {odd_sum}")
# Refactored version using list comprehensions
even_sum = sum(num for num in numbers_tuple if num % 2 == 0)
odd_sum = sum(num for num in numbers_tuple if num % 2 != 0)
print(f"Refactored - Sum of even numbers: {even_sum}")
print(f"Refactored - Sum of odd numbers: {odd_sum}")
```

Output:

Sum of even numbers: 290

Sum of odd numbers: 214
Refactored - Sum of even numbers: 290
Refactored - Sum of odd numbers: 214