Github Link: https://github.com/Kenche-Koushik/Tasks/tree/main/Day%201

Task 1: Install Jupyter Notebook / Google Colab.
Learn how to use them.
Writeup on what is its use.

Jupyter Notebook

A **notebook** is a shareable document that combines computer code, plain language descriptions, data, rich visualizations like 3D models, charts, graphs and figures, and interactive controls. A notebook, along with an editor (like JupyterLab), provides a fast interactive environment for prototyping and explaining code, exploring and visualizing data, and sharing ideas with others.

Jupyter Notebook is an open-source interactive computing environment that allows users to create and share documents containing:

- Live Code
- Mathematical equations
- Visualizations
- Explanatory text

It's widely used in **data science, machine learning, and research** because it makes experimentation easier.

Google Colab

Google Colab is a cloud-based version of Jupyter provided by Google. It requires no installation and provides free access to GPUs and TPUs, making it ideal for training deep learning models.

Key Uses

1. **Interactive Development:** Run code in small chunks, test ideas quickly.
2. **Data Analysis & Visualization:** Combine code + charts + explanations.
3. **Machine Learning Experiments:** Train/test models interactively.
4. **Collaboration & Sharing:** Share notebooks like Google Docs.
5. **Reproducibility:** Others can rerun your notebook and get the same results.

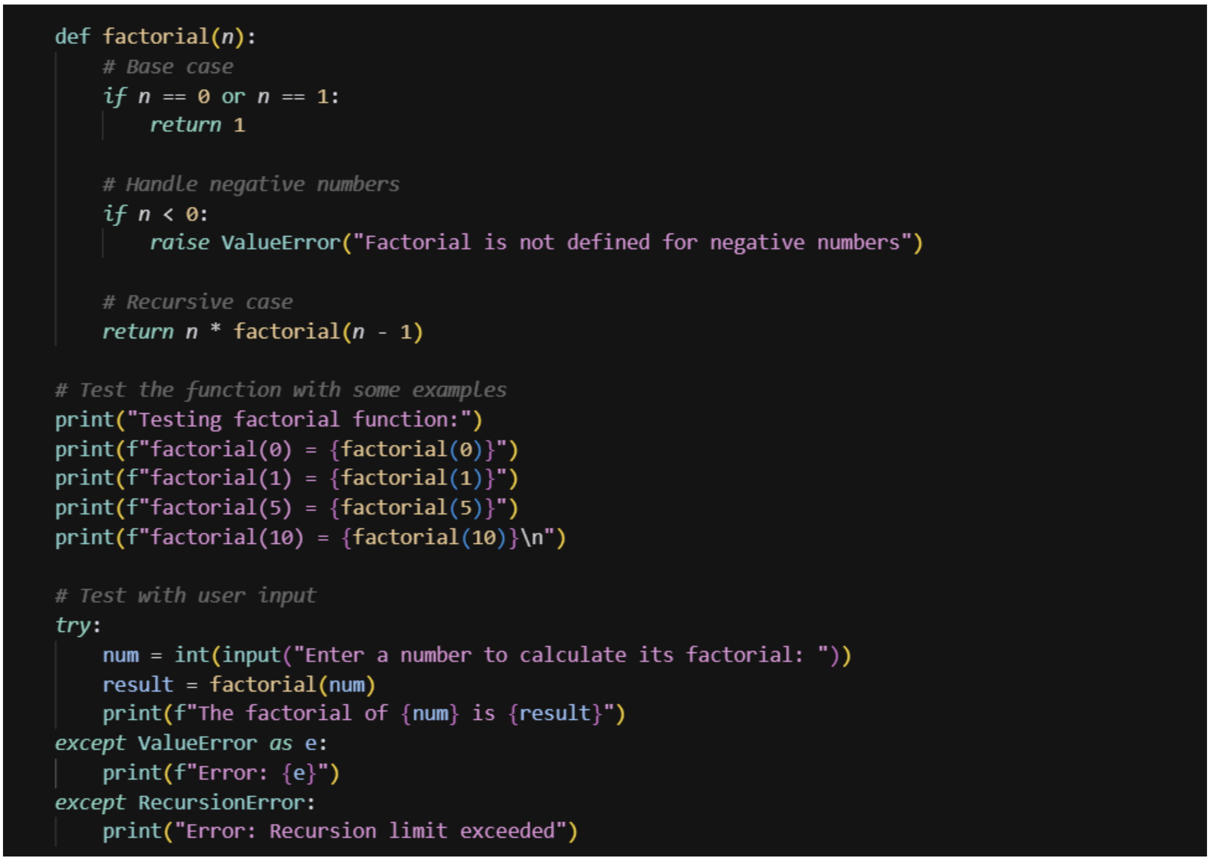
Task 3: Write function to calculate factorial of a number (do not use ChatGPT to generate the code)
Use recursive method for this. And give me pros and cons of using this method.

Features of the Implementation

- **Base Case Handling:** Returns 1 for 0! and 1!.
- **Negative Input Handling:** Raises an error if the input is negative.

- **Recursive Case:** Implements factorial using the mathematical definition.
- **Error Handling:** Catches ‘ValueError’ for invalid inputs and ‘RecursionError’ if recursion depth is exceeded.
- **User Interaction:** Accepts user input and computes factorial dynamically.

Implementation



```
def factorial(n):
    # Base case
    if n == 0 or n == 1:
        return 1

    # Handle negative numbers
    if n < 0:
        raise ValueError("Factorial is not defined for negative numbers")

    # Recursive case
    return n * factorial(n - 1)

# Test the function with some examples
print("Testing factorial function:")
print(f"factorial(0) = {factorial(0)}")
print(f"factorial(1) = {factorial(1)}")
print(f"factorial(5) = {factorial(5)}")
print(f"factorial(10) = {factorial(10)}\n")

# Test with user input
try:
    num = int(input("Enter a number to calculate its factorial: "))
    result = factorial(num)
    print(f"The factorial of {num} is {result}")
except ValueError as e:
    print(f"Error: {e}")
except RecursionError:
    print("Error: Recursion limit exceeded")
```

Pros of Recursive Method

1. **Simplicity:** The recursive approach closely matches the mathematical definition of factorial.
2. **Readability:** Code is concise and easy to understand.
3. **Good for learning:** Helps in understanding recursion, base cases, and function calls.

Cons of Recursive Method

1. **Inefficiency for Large Inputs** – Recursive calls consume extra memory and are slower compared to iteration.
2. **Recursion Depth Limit** – Python has a default recursion limit (~1000), so large inputs can cause a ‘RecursionError’.
3. **Stack Usage** – Each recursive call adds a new frame to the call stack, which may lead to stack overflow for very large numbers.

Note: The code is available in “Day_1.ipynb”

Task 4: Load a data set (csv) of students marks in a calls for multiple subjects (create a csv file using ChatGPT)-

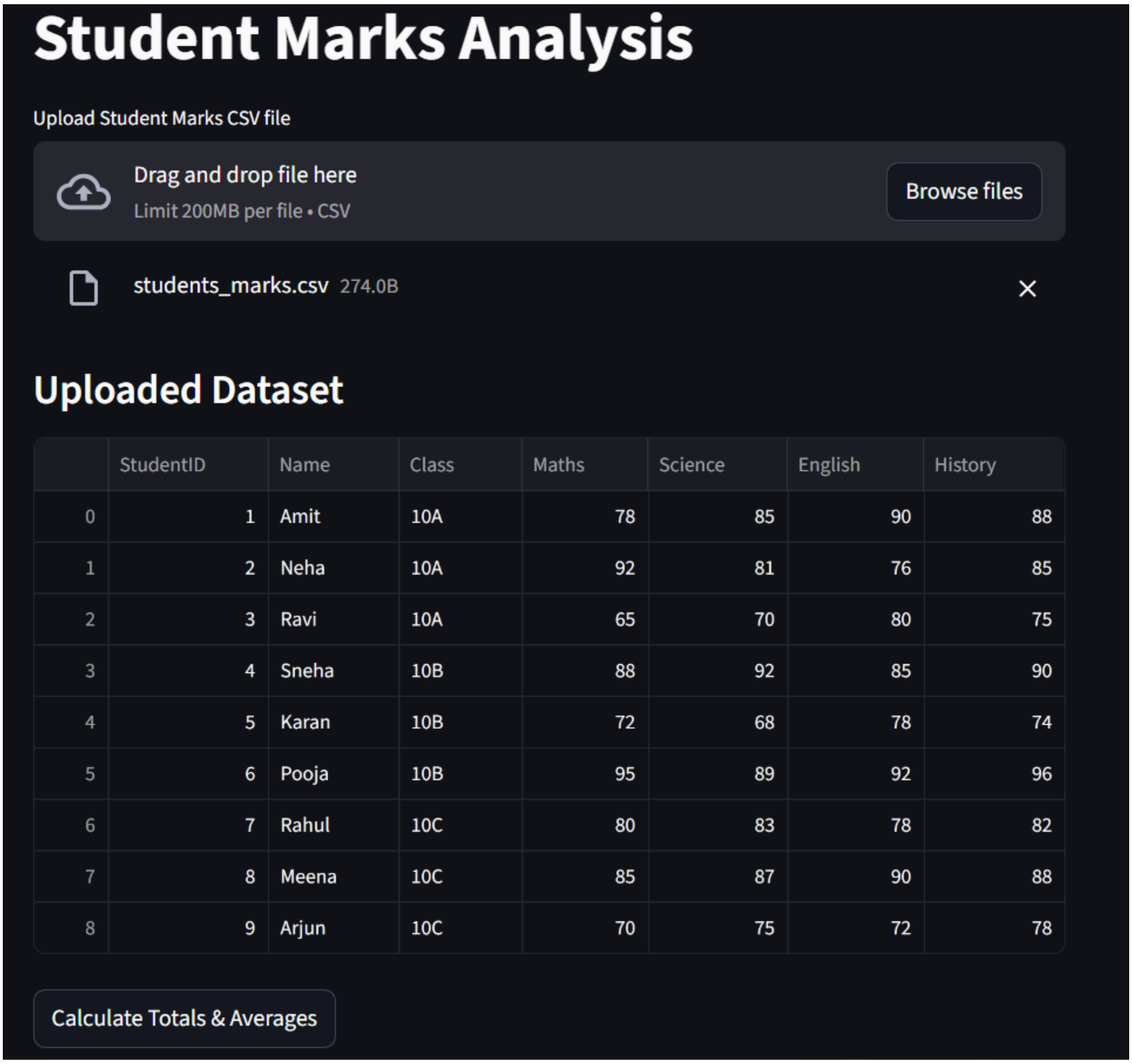
Find out total marks of each student. Find out average marks for each student. Find out class average in each subject.

Find out average marks for each class in each subject.

The code is available in “Day_1.ipynb”

Task 5: Create a streamlit application for above. (upload csv file, and show the data set, and button to calculate averages)

The is available in “app.py”



Student Marks Analysis

Upload Student Marks CSV file

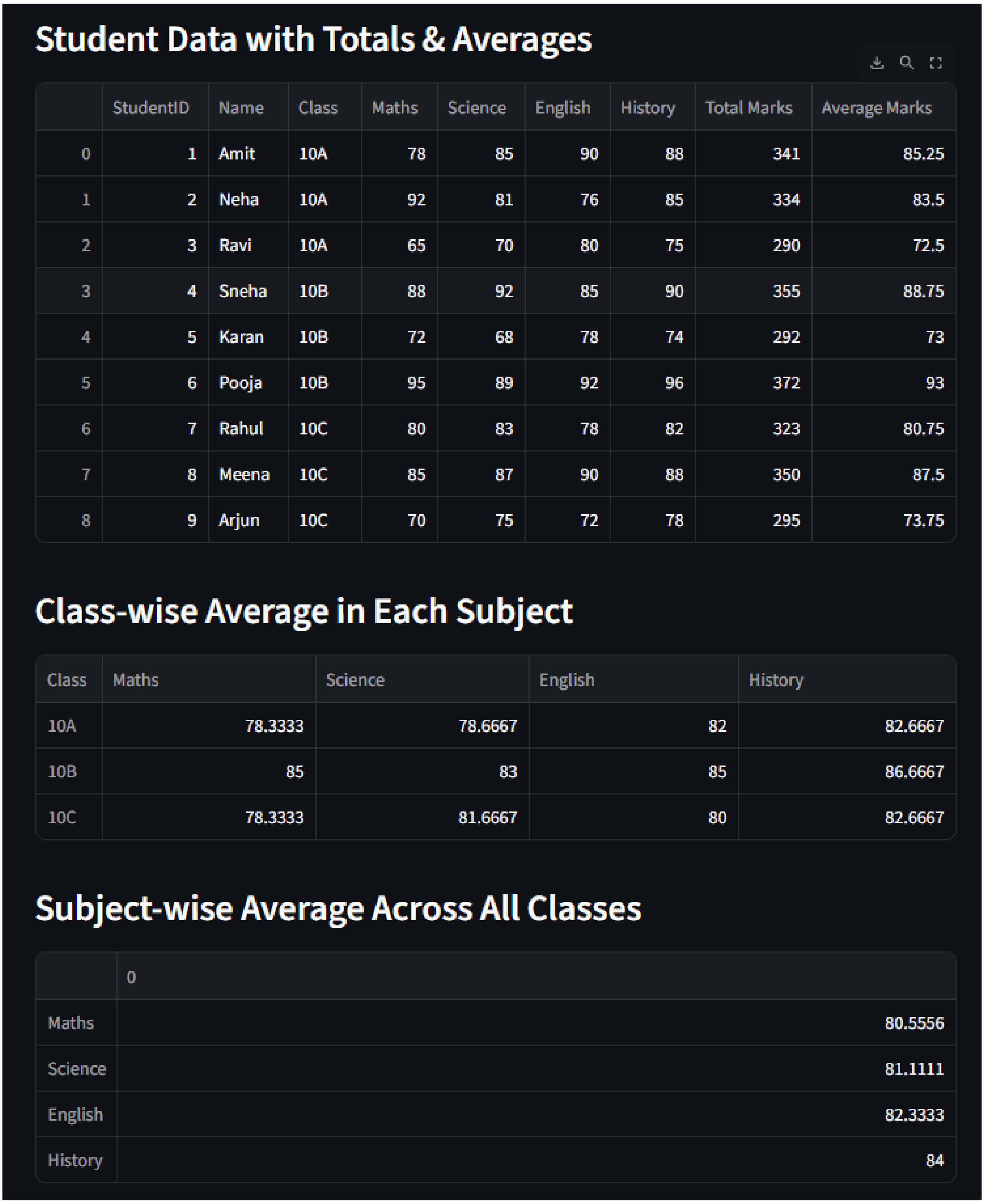
Drag and drop file here

Limit 200MB per file • CSV

students_marks.csv 274.0B

Uploaded Dataset

	StudentID	Name	Class	Maths	Science	English	History
0	1	Amit	10A	78	85	90	88
1	2	Neha	10A	92	81	76	85
2	3	Ravi	10A	65	70	80	75
3	4	Sneha	10B	88	92	85	90
4	5	Karan	10B	72	68	78	74
5	6	Pooja	10B	95	89	92	96
6	7	Rahul	10C	80	83	78	82
7	8	Meena	10C	85	87	90	88
8	9	Arjun	10C	70	75	72	78



Student Data with Totals & Averages

	StudentID	Name	Class	Maths	Science	English	History	Total Marks	Average Marks
0	1	Amit	10A	78	85	90	88	341	85.25
1	2	Neha	10A	92	81	76	85	334	83.5
2	3	Ravi	10A	65	70	80	75	290	72.5
3	4	Sneha	10B	88	92	85	90	355	88.75
4	5	Karan	10B	72	68	78	74	292	73
5	6	Pooja	10B	95	89	92	96	372	93
6	7	Rahul	10C	80	83	78	82	323	80.75
7	8	Meena	10C	85	87	90	88	350	87.5
8	9	Arjun	10C	70	75	72	78	295	73.75

Class-wise Average in Each Subject

Class	Maths	Science	English	History
10A	78.3333	78.6667	82	82.6667
10B	85	83	85	86.6667
10C	78.3333	81.6667	80	82.6667

Subject-wise Average Across All Classes

	0
Maths	80.5556
Science	81.1111
English	82.3333
History	84

Task 6: Convert an uploaded CSV file into JSON format.

The code is available in “Day_1.ipynb”