**Task #1 – Zero-Shot Prompting with Conditional Validation**

Objective

Use zero-shot prompting to instruct an AI tool to generate a function that validates an Indian mobile number.

Requirements

* The function must ensure the mobile number:
  + Starts with 6, 7, 8, or 9
  + Contains exactly 10 digits

Expected Output

* A valid Python function that performs all required validations without using any input-output examples in the prompt.



**Task #2 – One-Shot Prompting with Edge Case Handling**

Objective

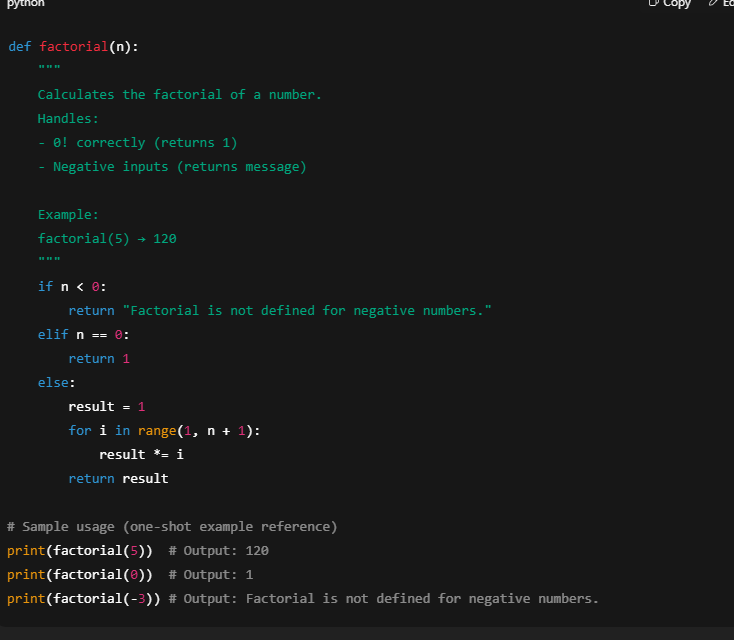
Use one-shot prompting to generate a Python function that calculates the factorial of a number.

Requirements

* Provide one sample input-output pair in the prompt to guide the AI.
* The function should handle:
  + 0! correctly
  + Negative input by returning an appropriate message

Expected Output

* A Python function with correct factorial logic and edge case handling, generated from a single example.



**Task #3 – Few-Shot Prompting for Nested Dictionary Extraction**

Objective

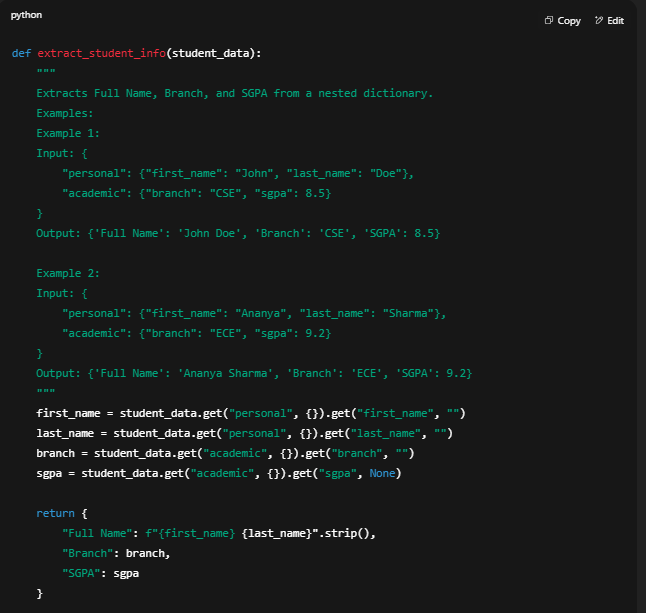
Use few-shot prompting (2–3 examples) to instruct the AI to create a function that parses a nested dictionary representing student information.

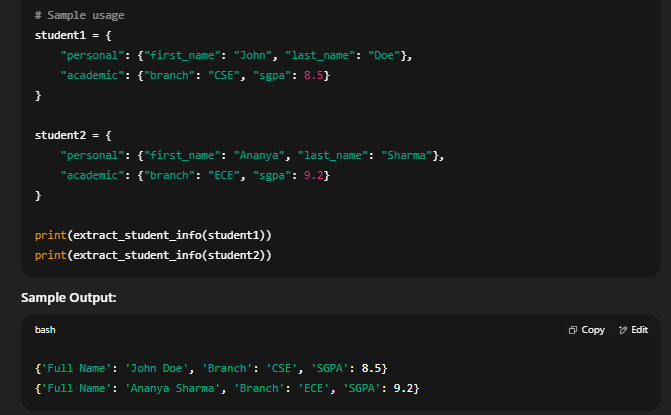
Requirements

* The function should extract and return:
  + Full Name
  + Branch
  + SGPA

Expected Output

A reusable Python function that correctly navigates and extracts values from nested dictionaries based on the provided examples





**Task #4 – Comparing Prompting Styles for File Analysis**

Objective

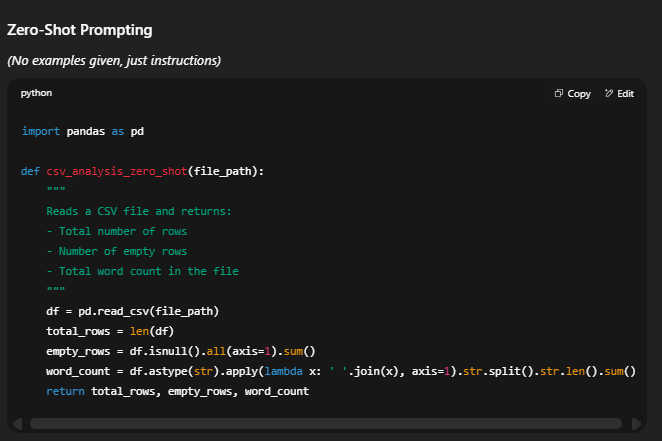
Experiment with zero-shot, one-shot, and few-shot prompting to generate functions for CSV file analysis.

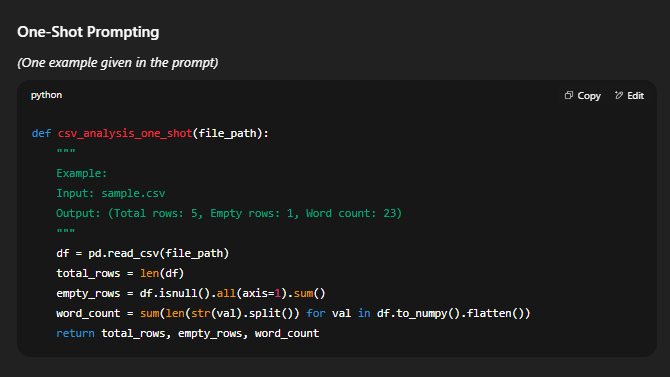
Requirements

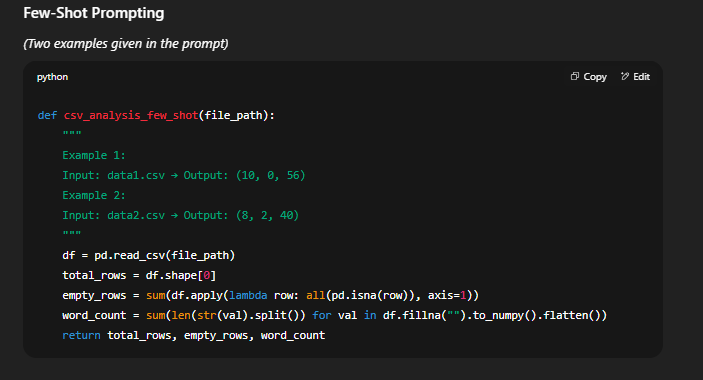
* Each generated function should:
  + Read a .csv file
  + Return the total number of rows
  + Count the number of empty rows
  + Count the number of words across the file

Expected Output

* Working Python functions for each prompting style, with a brief reflection comparing their accuracy, clarity, and efficiency.







**Task #5 – Few-Shot Prompting for Text Processing and Word** **Frequency**

Objective

Use few-shot prompting (with at least 3 examples) to generate a Python function that processes text and analyzes word frequency.

Requirements

The function must:

* Accept a paragraph as input
* Convert all text to lowercase
* Remove punctuation
* Return the most frequently used word

Expected Output

* A functional Python script that performs text cleaning, tokenization, and returns the most common word using only the examples provided in the prompt

**Note:** Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots

