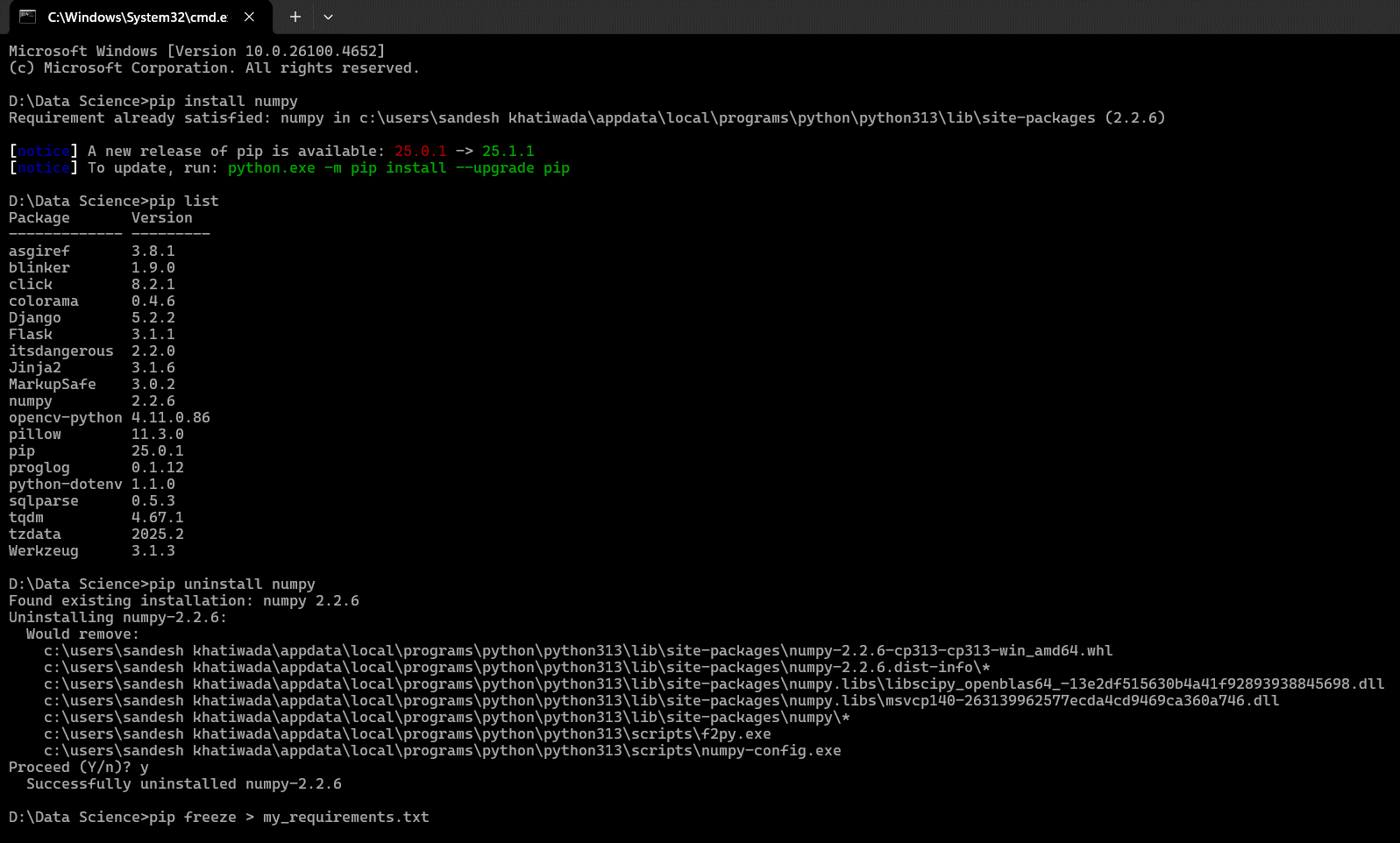
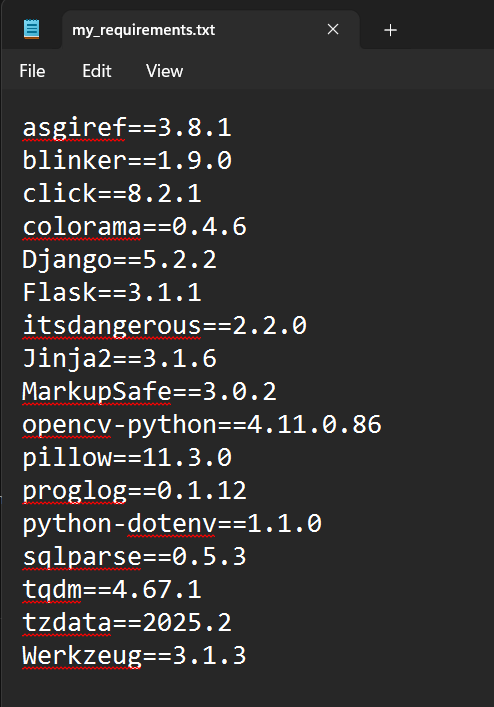
**Exercise 1: Python Package Management**

1. Using pip:

* Open your terminal or command prompt.
* Install the numpy package.
* List all currently installed packages.
* Uninstall the numpy package.
* If you have any packages installed, generate a requirements.txt file that lists

them. Open the my\_requirements.txt file and review its contents.



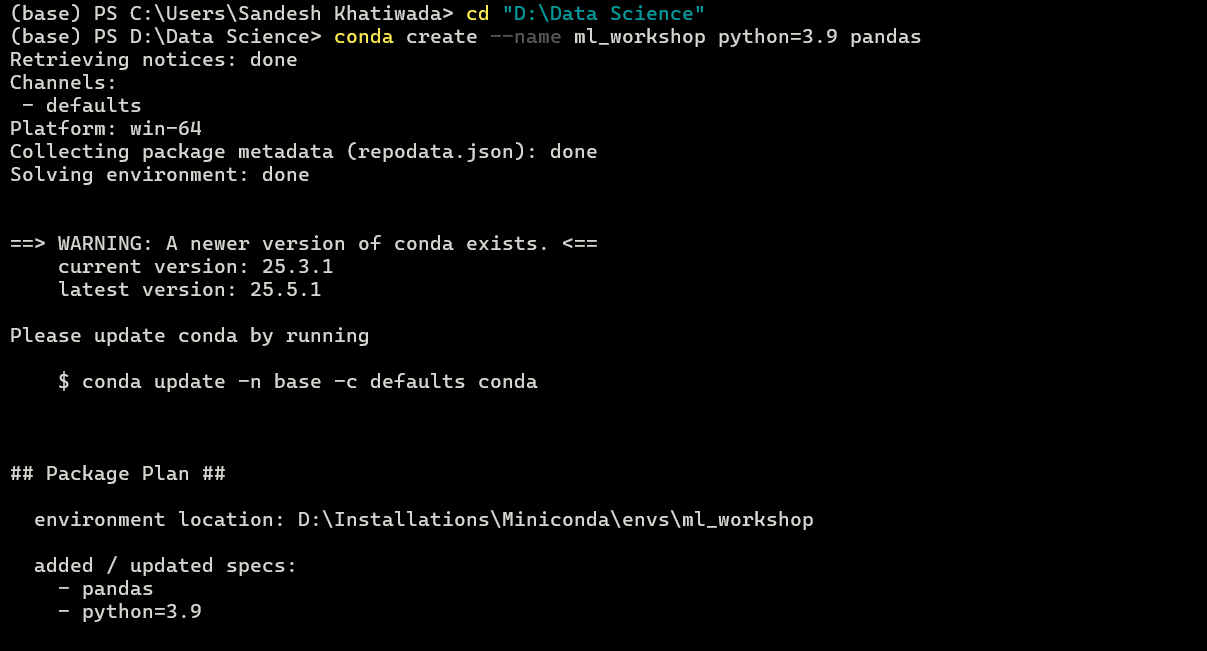
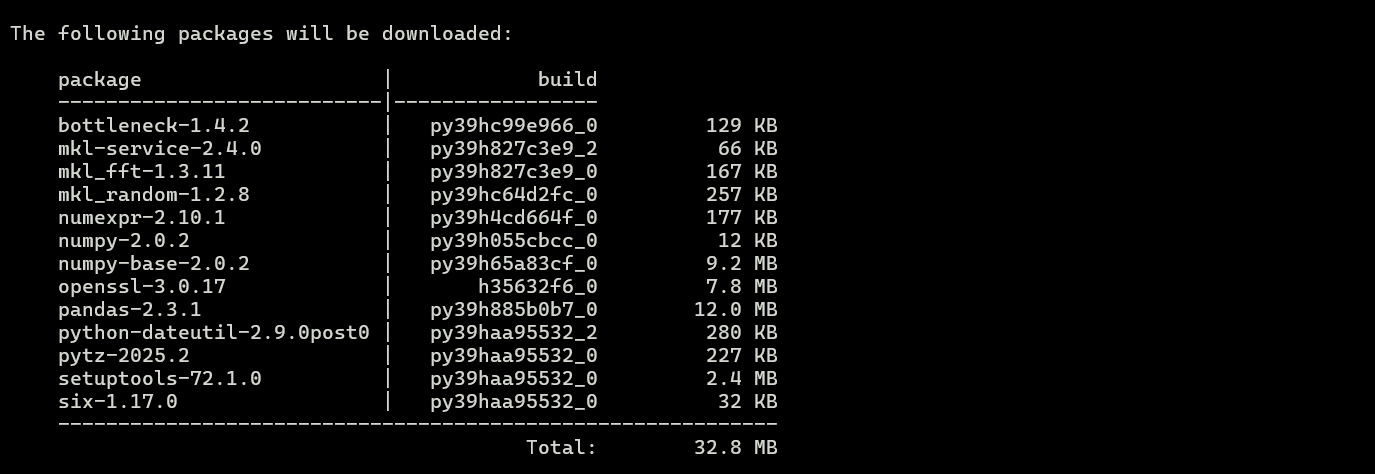
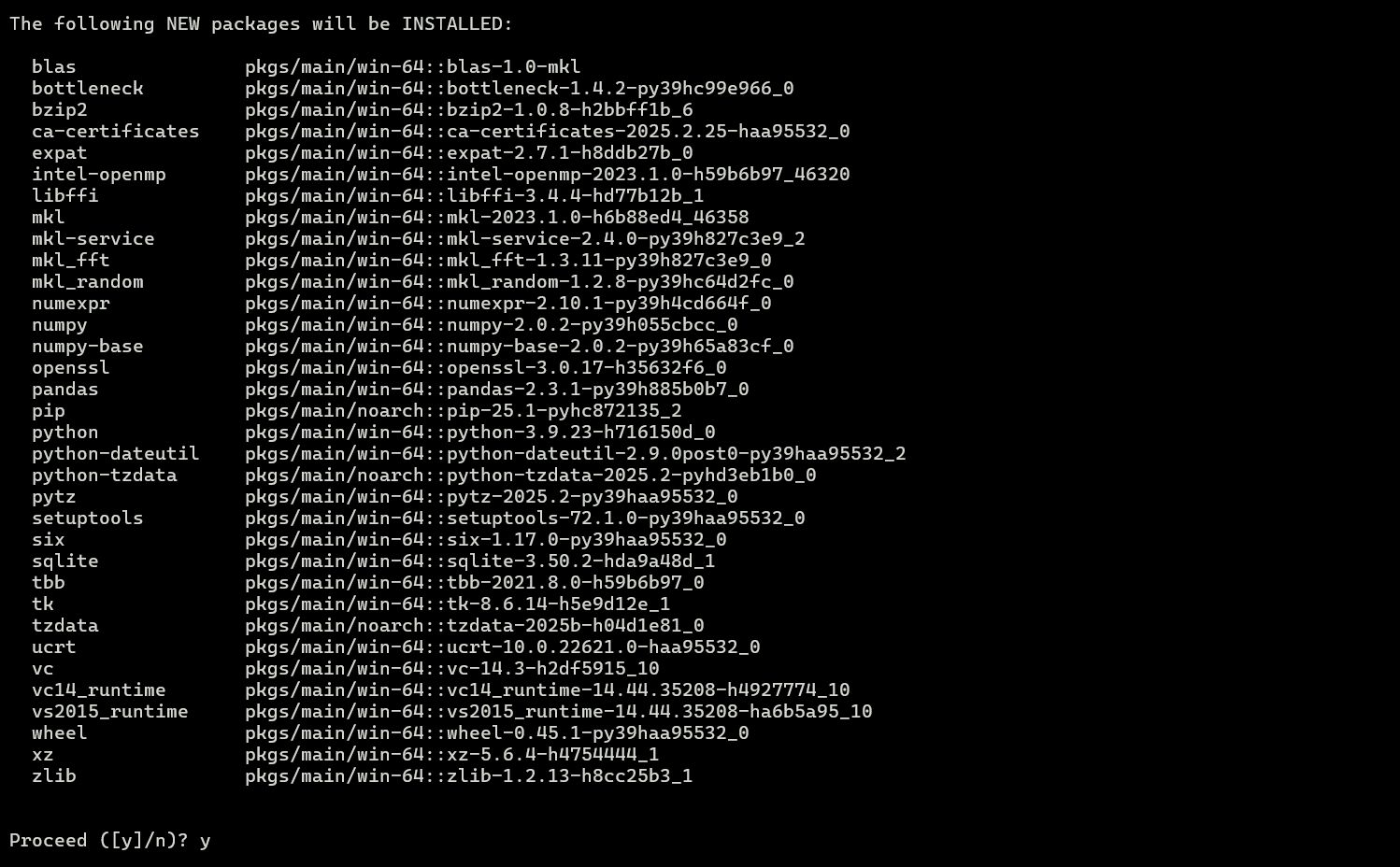


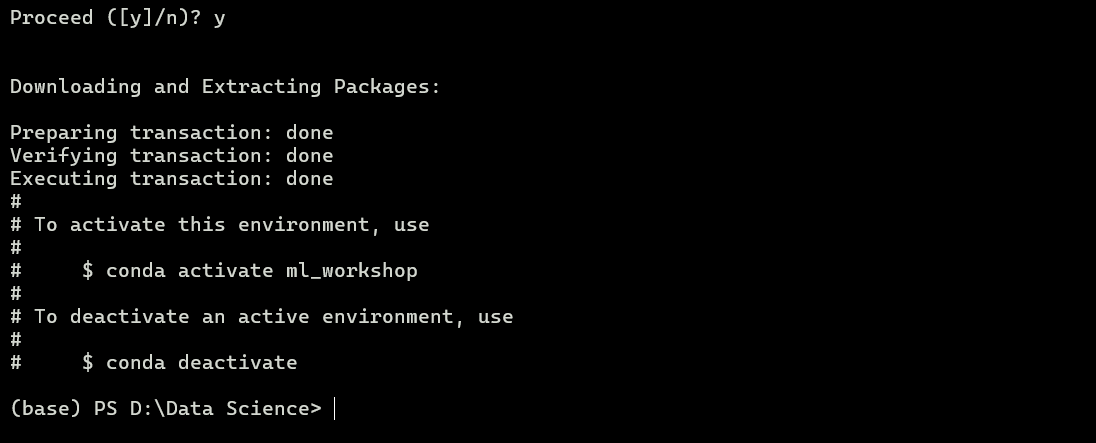
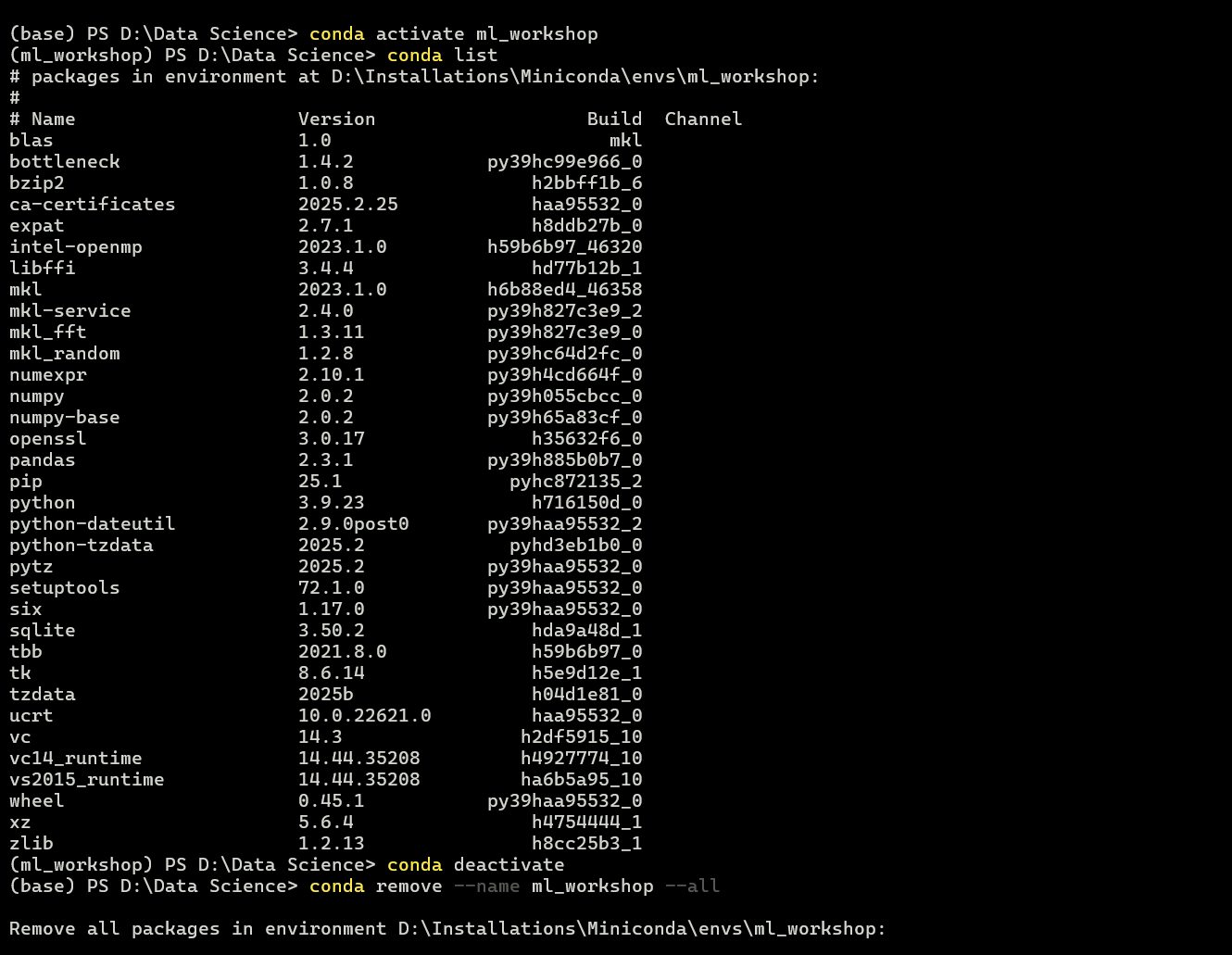
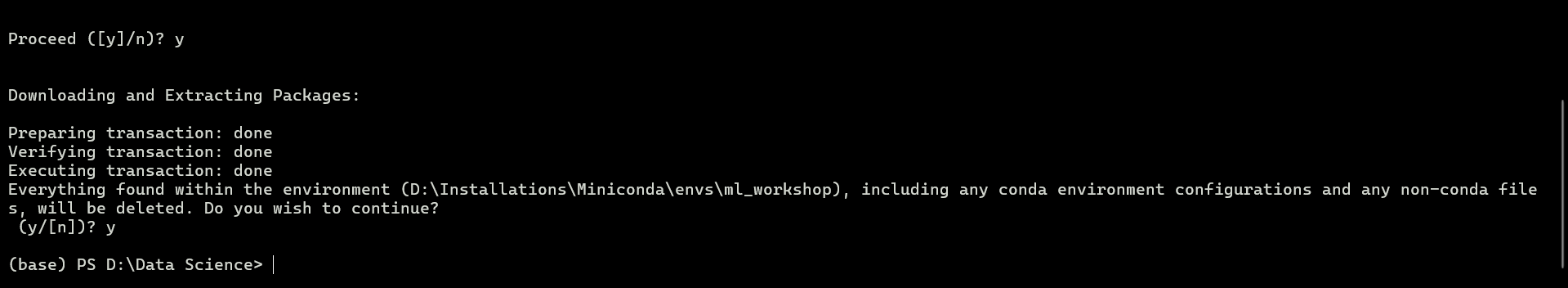
2. Using conda (if Anaconda/Miniconda is installed):

* Open your Anaconda Prompt or terminal.
* Create a new conda environment named ml\_workshop with Python 3.9 and the

pandas library.

* Activate the ml\_workshop environment.
* List all packages within the activated ml\_workshop environment.
* Deactivate the ml\_workshop environment.
* (Optional) Remove the ml\_workshop environment.



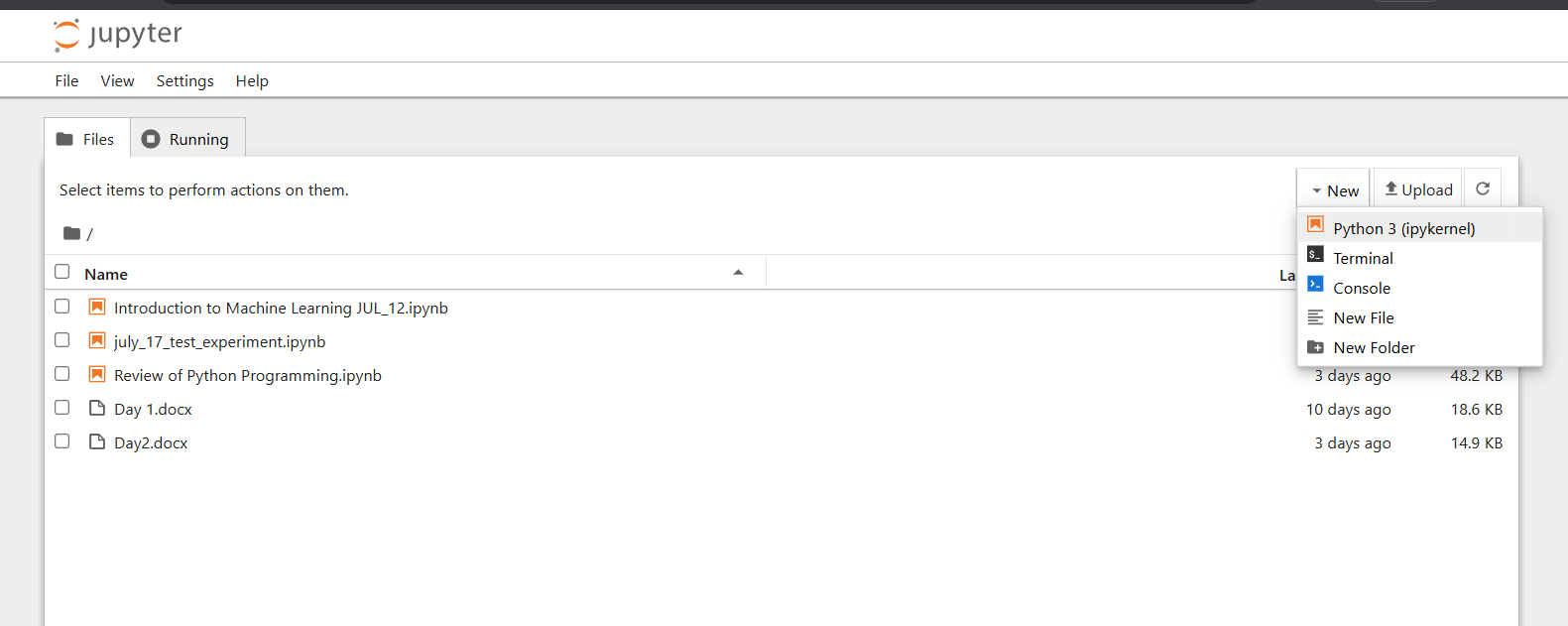


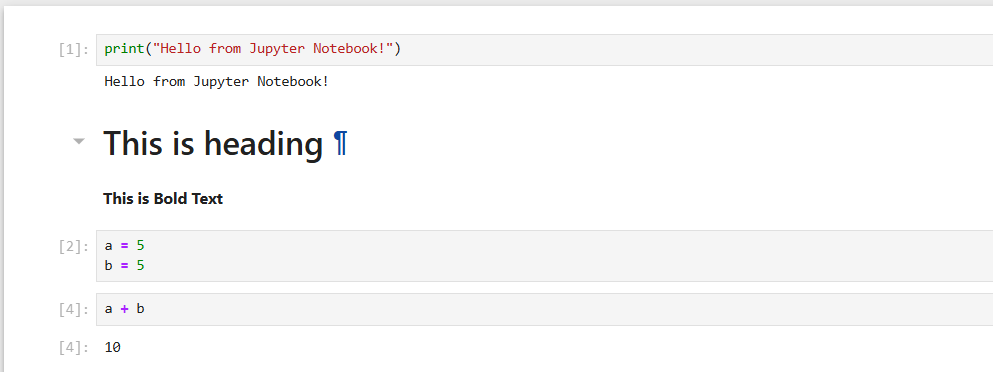
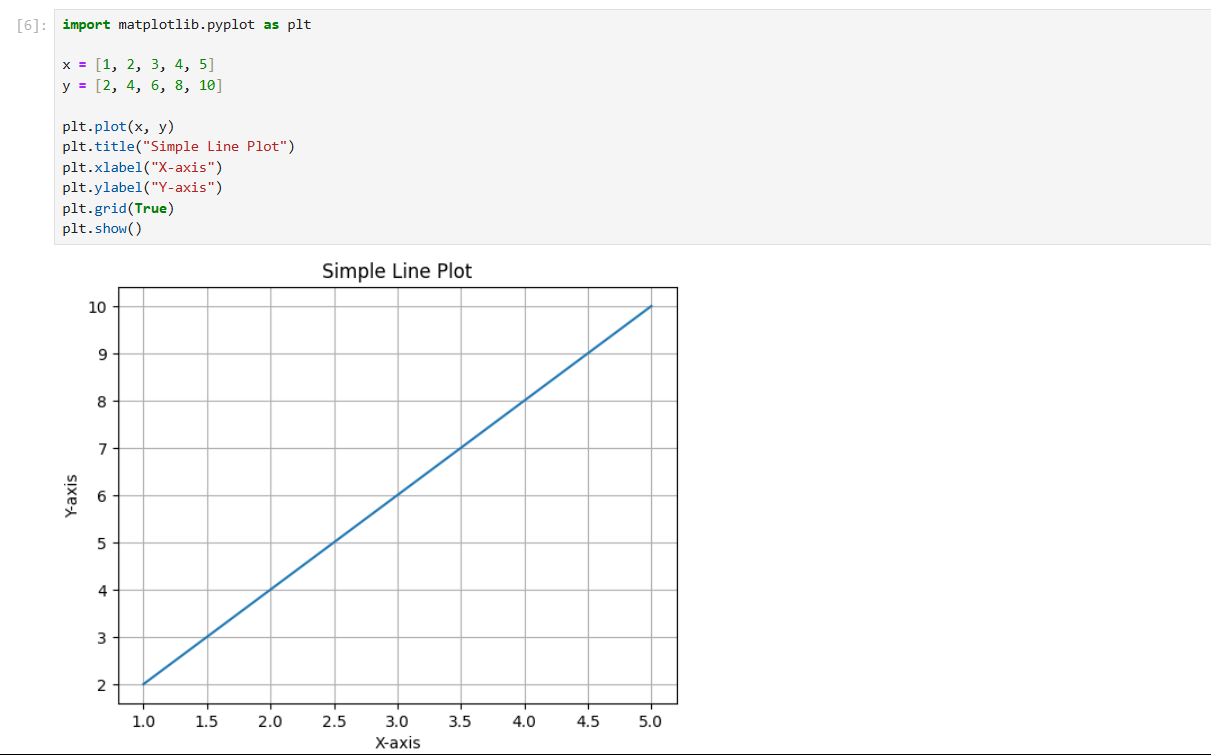
**Exercise 2: Jupyter Notebook and Google Colab Exploration**

1. Jupyter Notebook (Local):

* Launch Jupyter Notebook from your terminal.
* Create a new Python 3 notebook.
* In the first code cell, print the message "Hello from Jupyter Notebook!".
* Change a new cell's type to Markdown and add a heading "My First Jupyter
* Notebook" along with some bold text.
* In a new code cell, perform a simple addition of two numbers and print the result.
* In another code cell, import the matplotlib.pyplot library and create a simple
* line plot.
* Save your notebook and close the Jupyter server.

Launching Jupyter Notebook:



Simple Notebook Works:

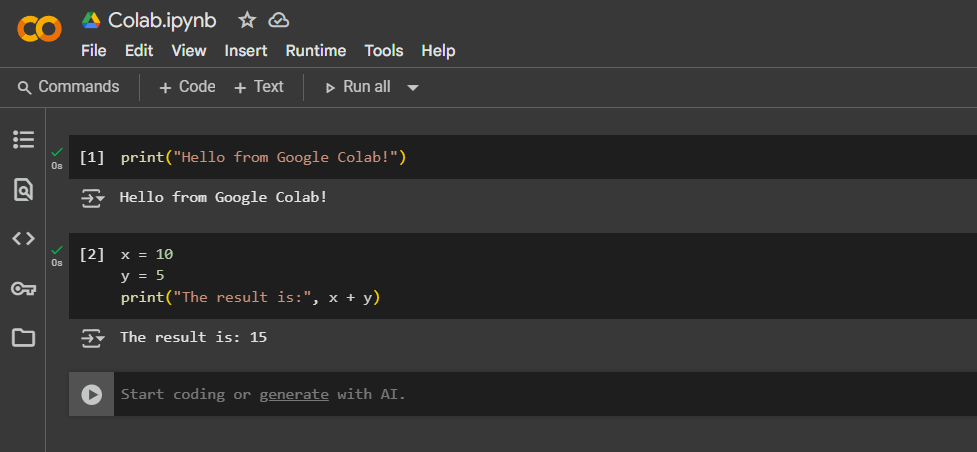
2. Google Colab:

* Access Google Colaboratory (colab.research.google.com).
* Create a new notebook.
* Repeat the "Hello world" and simple calculation steps from the Jupyter Notebook
* exercise in Colab code cells.
* (Optional) Change the runtime type to GPU/TPU and run a command to display GPU information.
* Install a temporary package (e.g., tensorflow-datasets) using a shell command

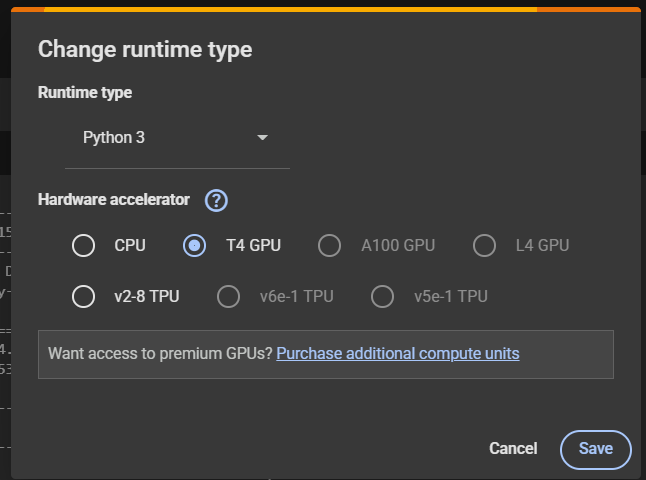
within a code cell.

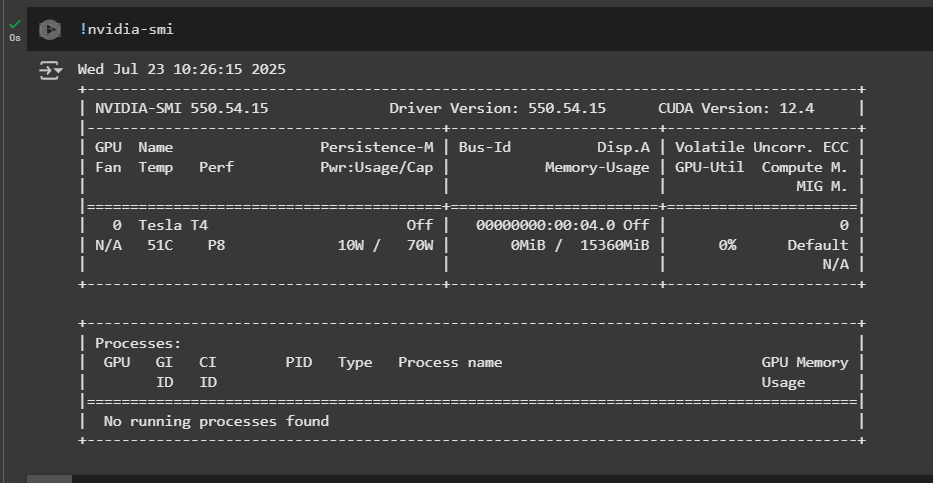
* Explore the options to share and save a copy of your notebook.

Access Google Colab

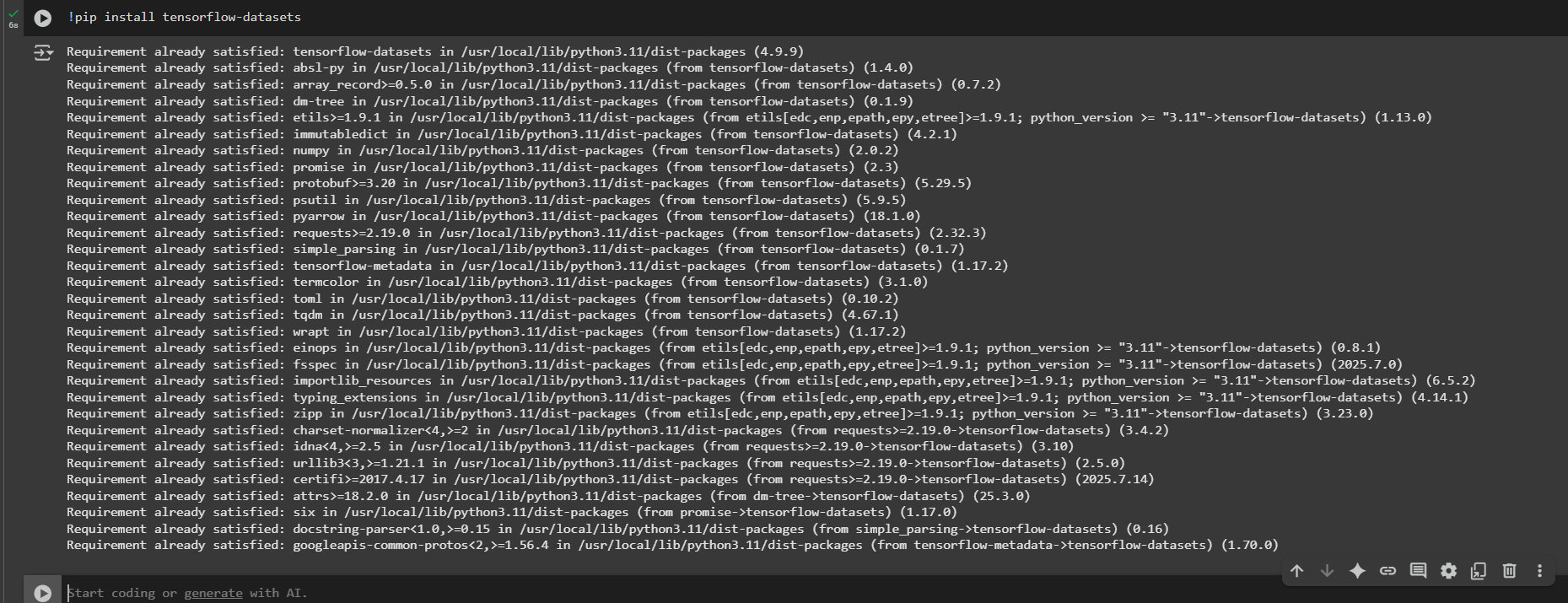


Change Runtime Type:

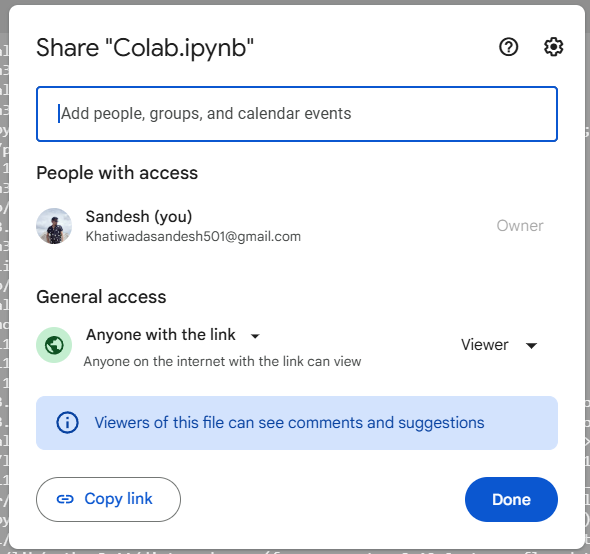




Temporary Package:



Share Option:



**Exercise 3: Machine Learning Concepts - Mini-Quiz & Discussion**

Instructions: Answer the following questions. Be prepared to discuss your reasoning.

1. Email Spam Detection: You are building a system to identify whether an incoming

email is spam or not.

* Is this a Supervised or Unsupervised learning problem? Justify your answer with a key characteristic of the chosen type.

Ans: The model is trained on labeled data, where each email is already marked as "spam" or "not spam". This is the key characteristic of supervised learning i.e. using input-output pairs for training.

* What specific type of supervised learning task is it (Classification or Regression)?

Ans: It is classification as it distinguishes between spam and ham.

2. Customer Segmentation: You have a dataset of customer purchase behaviors (e.g.,

items bought, frequency, total spending) and want to group similar customers together to identify distinct market segments for targeted marketing campaigns.

* Is this a Supervised or Unsupervised learning problem? Explain your choice.

Ans: It is unsupervised learning as There are no predefined labels, the goal is to discover patterns or natural groupings in the customer behavior data without explicit instruction.

* What common unsupervised learning technique would you likely use for this task, and what is its primary goal?

Ans: We use K-Means Clustering to partition customers into distinct groups (clusters) based on similarity in their purchase behavior. This helps businesses target each segment effectively.

3. pip vs. conda:

* What is the fundamental difference in what pip and conda manage?

Ans: pip is a Python package manager, it installs only Python packages from the Python Package Index (PyPI).  
conda is a package, environment, and dependency manager it can install both Python and non-Python dependencies (like C libraries or compilers) and manage environments.

* Provide a scenario where conda would be a more advantageous choice over pip for managing a project's dependencies.

Ans: If you're working on a scientific project involving heavy libraries like numpy, opencv, tensorflow, or non-Python dependencies (e.g., CUDA for GPU), conda is better because:

* + It handles binary dependencies better
  + It avoids common version conflicts
  + It can install non-Python tools along with Python packages

4. Neural Networks and Activation Functions:

* True or False: Neural Networks are a type of machine learning model where

interconnected "neurons" process information in layers, and each neuron typically applies an activation function to its output.

Ans: True

* Briefly explain the purpose of an activation function within a neural network. Why is it important for a neural network to use non-linear activation functions?

Ans: Activation functions introduce non-linearity to a neural network.

* 1. Without non-linear activation functions, a neural network would behave like a linear model, regardless of how many layers it has.
  2. Non-linear functions (like ReLU, Sigmoid, Tanh) allow the network to learn complex patterns, approximate any function, and solve non-linear problems such as image recognition, NLP, etc.