Short Answer Questions

Q1: Explain the primary differences between TensorFlow and PyTorch. When would you choose one over the other?

Differences:

Feature	TensorFlow	PyTorch
Computation graph	Uses *static* computation graphs (defined before execution).	Uses *dynamic* computation graphs (built on the fly during execution).
Ease of debugging	Harder to debug due to static graphs.	Easier to debug with Pythonic style and dynamic nature.
Deployment	Strong deployment tools (TensorFlow Serving, TensorFlow Lite).	Deployment improving through TorchServe but slightly less mature.
Ecosystem	Has TensorBoard for visualization and Keras for simplified API.	Integrates well with Python tools and supports research flexibility.

When to choose:

- ✅ **TensorFlow** – when building **production-ready** AI systems (e.g., mobile apps, enterprise solutions) needing **easy deployment** and **visualization** tools.
- ✅ **PyTorch** – when doing **AI research, prototyping**, or **experimentation** because of its **flexibility and simplicity**.

Q2: Describe two use cases for Jupyter Notebooks in AI development.

1. **Data Exploration and Visualization:**
   Jupyter allows developers to interactively load, clean, and visualize datasets using tools like pandas and matplotlib — making it perfect for **exploratory data analysis (EDA)**.

2. **Model Prototyping and Experimentation:**
   It provides an interactive environment to test different **machine learning models, parameters, and hyperparameters**, showing real-time outputs without needing a full application setup.

Q3: How does spaCy enhance NLP tasks compared to basic Python string operations?

- **spaCy** provides **pre-trained language models** and advanced features like **tokenization, part-of-speech tagging, named entity recognition (NER), and dependency parsing** — all of which are **context-aware**.

- Basic Python string operations (like `.split()` or `.replace()`) only manipulate raw text without understanding **grammar or meaning**.

- Example: spaCy can recognize `"John went to Kenya"` → *John* = PERSON, *Kenya* = LOCATION — which basic string operations cannot do.

---

2. Comparative Analysis

Feature	Scikit-learn	TensorFlow
Target applications	Classical Machine Learning (e.g., regression, classification, clustering).	Deep Learning (e.g., neural networks, computer vision, NLP).
Ease of use for beginners	Very beginner-friendly; simple, consistent APIs.	Steeper learning curve; requires understanding of tensors and computational graphs.
Community support	Strong community in academia and traditional ML users.	Huge community in deep learning and production AI, supported by Google.

✅ **Summary:**

- Use **Scikit-learn** for **simpler models** and **quick experiments**.

- Use **TensorFlow** for **complex deep learning** and **large-scale neural networks**.