PART 1;THEORETICAL UNDERSTANDING

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

Answer:
TensorFlow and PyTorch are both deep learning frameworks, but they differ mainly in how they handle computation graphs and usability.

- **TensorFlow** uses *static computation graphs* (you define the model first, then run it). It's preferred for **production deployment**, mobile and web integration, and large-scale projects because it works well with **TensorFlow Serving** and **TensorFlow Lite**.

- **PyTorch** uses *dynamic computation graphs* (you can modify the model as it runs), making it very flexible and easy to debug. It's preferred in **research and experimentation** because of its **Pythonic** style and real-time debugging.

  Use **PyTorch** for **research & experimentation**, and **TensorFlow** for **production & scalability**.

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

Answer:

1. **Prototyping and experimentation:** Developers can test code, visualize results, and modify models interactively.

2. **Documentation and reporting:** Jupyter combines code, visualizations, and markdown text, making it ideal for creating tutorials, research papers, and presentations.

Example: Training a small model on sample data and showing graphs of accuracy over epochs.

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

Answer:
spaCy is a specialized NLP library designed for advanced text processing. Unlike basic string operations that only handle raw text, spaCy provides:

- **Tokenization, lemmatization, and POS tagging** (understanding grammatical structure)

- **Named Entity Recognition (NER)** to identify names, brands, and locations
- **Dependency parsing** to understand sentence meaning

spaCy understands *language structure* not just text patterns.

Comparative Analysis: Scikit-learn vs TensorFlow

Feature	Scikit-learn	TensorFlow
Target Applications	Classical ML (SVM, Decision Trees, Regression, Clustering)	Deep Learning (CNNs, RNNs, Transformers)
Ease of Use (Beginners)	Easier to learn; simpler syntax and smaller datasets	Steeper learning curve; more setup required
Community Support	Very strong for classical ML and academic use	Huge community; widely used in production and industry