Short Answer Questions

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

A1: TensorFlow and PyTorch are both popular deep learning frameworks. TensorFlow, developed by Google, uses static computation graphs (though it supports dynamic graphs via tf.function in newer versions), making it better suited for production environments. PyTorch, developed by Facebook, uses dynamic computation graphs, offering more flexibility and ease of debugging. Choose PyTorch for research and experimentation due to its intuitive nature, and TensorFlow for deploying scalable production models, especially with TensorFlow Serving and TensorFlow Lite.

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

A2:

1. **Interactive Experimentation**: Jupyter Notebooks allow developers to write and run code in chunks, visualize data, and see results immediately. This is useful for model prototyping and debugging.
2. **Documentation and Sharing**: They support markdown, code, and outputs in one file, making them ideal for documenting experiments and sharing work with collaborators or for educational purposes.

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

A3: spaCy is a powerful NLP library that provides pre-trained models and optimized pipelines for tokenization, POS tagging, named entity recognition, and more. Unlike basic string operations, which rely on manual parsing and pattern matching, spaCy understands linguistic structure, handles edge cases better, and is significantly faster and more accurate for real-world text processing.

Comparative Analysis

Compare Scikit-learn and TensorFlow in terms of:

- Target Applications: Scikit-learn is best for classical machine learning tasks (e.g., linear regression, decision trees, clustering), while TensorFlow excels in deep learning tasks (e.g., neural networks, image and speech recognition).
- Ease of Use for Beginners: Scikit-learn is generally easier for beginners due to its consistent API, straightforward syntax, and excellent documentation. TensorFlow has a steeper learning curve, though TensorFlow 2.x has improved ease of use with Keras integration.