Part 1: Theoretical Understanding

1. Short Answer Questions

Q1: Explain the primary differences between TensorFlow and PyTorch. When would you choose one over the other?
TensorFlow is optimized for large-scale deployment with static computation graphs and strong support for production, mobile, and edge deployment. PyTorch uses dynamic computation graphs, making it more intuitive and flexible for experimentation and research. TensorFlow is best suited for enterprise and production environments, while PyTorch is preferred for rapid prototyping and academic use.

Q2: Describe two use cases for Jupyter Notebooks in AI development.
1. Model experimentation: Jupyter Notebooks allow developers to test algorithms, visualize data, and tweak models interactively.
2. Education and collaboration: They are ideal for teaching AI concepts and sharing code with explanations and outputs in one interactive environment.

Q3: How does spaCy enhance NLP tasks compared to basic Python string operations?
spaCy provides advanced linguistic features such as tokenization, lemmatization, part-of-speech tagging, and named entity recognition. Unlike simple Python string operations, it understands linguistic structure and context, enabling more accurate and efficient natural language processing.

2. Comparative Analysis

Target Applications: Scikit-learn focuses on classical machine learning algorithms such as regression, classification, and clustering. TensorFlow specializes in deep learning and neural networks.

Ease of Use for Beginners: Scikit-learn is easier to learn and use due to its consistent API and straightforward workflow. TensorFlow has a steeper learning curve but offers greater flexibility for complex models.

Community Support: Both have large, active communities. Scikit-learn is popular in academic and traditional ML research circles, while TensorFlow has stronger support in production-level and enterprise AI development.