

Part 1: Theoretical Understanding (40%)

1. Short Answer Questions

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

Answer:

TensorFlow uses a static computation graph, while PyTorch uses a dynamic computation graph.

PyTorch is more intuitive and better for experimentation and debugging.

TensorFlow is more scalable and production-ready, especially for deploying on mobile and web.

Use PyTorch for research and fast prototyping.

Use TensorFlow for large-scale deployment and integration.

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

Answer:

1. Exploratory Data Analysis (EDA): Enables interactive data cleaning, visualization, and transformation.

2. Model Prototyping: Allows step-by-step model building with real-time outputs and debugging.

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

Answer:

spaCy uses pre-trained models to handle complex language tasks like tokenization, POS tagging, and NER.

It goes beyond string operations by understanding linguistic context and grammar, making NLP workflows faster and more accurate.

2. Comparative Analysis: Scikit-learn vs TensorFlow

Feature	Scikit-learn	TensorFlow
Target Applications	Classical ML (e.g., regression, clustering)	Deep Learning (e.g., CNNs, RNNs, DNNs)
Ease of Use	Beginner-friendly, simple API	Steeper learning curve, more configuration needed
Community Support	Strong for traditional ML workflows	Massive, with production/deployment tools (e.g., TF Lite)