Part 1 — Theoretical Understanding (40%)

Q1: Primary differences between TensorFlow and PyTorch.

Key differences

Computation graph style

PyTorch: dynamic, define-by-run (builds graph as code runs). Very intuitive for Pythonic debugging and for research/prototyping.

TensorFlow (historically): static graph (define-then-run). TensorFlow 2.x uses eager execution by default, which behaves more like PyTorch but still has tf.function for graph compilation.

API style

PyTorch: low-level, Pythonic, great for custom research models.

TensorFlow: higher-level ecosystem (Keras API), extensive tooling for production (TF Serving, TFLite).

Ecosystem

PyTorch: strong research community and rapid algorithm adoption.

TensorFlow: broad production tooling, deployment options, and enterprise integrations.

Deployment

TensorFlow: easier conversions to mobile/embedded (TFLite), TF Serving and TF Hub.

PyTorch: now strong too (TorchScript, TorchServe), but historically research-first.

When to choose which

Use PyTorch if you want fast experimentation, custom layers, or are doing research/prototyping.

Use TensorFlow (Keras) if you plan end-to-end production deployment, need mobile/edge support, or prefer a higher-level API.

Q2: Two use cases for Jupyter Notebooks in AI development

Exploratory Data Analysis (EDA): interactive plots, inline tables, iteratively cleaning and visualizing dataset distributions.

Prototyping & Demonstration: teach a concept, step through model-building, show training curves and example predictions inline; excellent for sharing reproducible experiments.

Q3: How spaCy enhances NLP tasks vs. basic Python string operations

Accurate tokenization (handles punctuation, contractions, languages).

Linguistic features: POS tagging, dependency parsing, named entities (NER) — all pre-trained and fast.

Pipeline & speed: vectorized, optimized Cython implementation; much faster and more robust than ad-hoc regex/string parsing.

Reusable pipeline components and easy integration for production (serialization, custom components).