PART 1 — THEORETICAL UNDERSTANDING (40%)

Q1: TensorFlow vs PyTorch

Feature	TensorFlow	PyTorch
Execution mode	Graph-based (static computation graph; better for deployment)	Eager execution (dynamic graph; easier debugging)
Ease of use	Requires more boilerplate; high-level API (Keras) simplifies it	More Pythonic; simple, intuitive syntax
Performance	Highly optimized for production (TensorFlow Serving, TF Lite)	Excellent for research and prototyping
Visualization	TensorBoard for metrics and graph visualization	TorchBoard and third-party tools exist but less integrated
When to choose	Choose TensorFlow for production-ready or mobile/embedded deployment	Choose PyTorch for fast prototyping, research, or flexibility

Q2: Use cases for Jupyter Notebooks in AI

1. **Interactive experimentation:** Ideal for trying out ML algorithms, visualizing intermediate results, and quick iteration.

2. **Reproducible research & teaching:** You can share notebooks with code, visualizations, and explanations in one file for tutorials, assignments, or demos.

Q3: spaCy vs basic Python string operations

Feature	spaCy	Basic Python
Tokenization	Context-aware (handles punctuation, contractions)	Manual `.split()` — inaccurate for language structures
Named Entity Recognition (NER)	Built-in ML models to detect entities like people, organizations, products	Requires manual pattern matching

POS tagging, dependency parsing	Automatic linguistic annotations	Not available
Speed & optimization	Cython backend → very fast	Slower for large text datasets

spaCy enables advanced NLP pipelines, while string ops are only for simple tasks.

Comparative Analysis: Scikit-learn vs TensorFlow

Criteria	Scikit-learn	TensorFlow
Target applications	Classical ML (SVM, decision trees, regression, clustering)	Deep learning (neural networks, CNNs, RNNs)
Ease of use	Easier for beginners — consistent API (`fit`, `predict`)	Steeper learning curve; more setup required
Community support	Very large for classical ML	Massive for DL + production
Best use case	Small tabular data, quick models	Large-scale data, images, NLP, deployment