🎓 📝 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?

TensorFlow and PyTorch are both prominent deep learning frameworks widely used in academia and industry.

- **TensorFlow**, developed by Google Brain, originally used a static computation graph (TF 1.x), requiring users to define the entire graph before execution. Since TensorFlow 2.x, it supports eager execution by default, making development more flexible while retaining robust deployment capabilities (TensorFlow Serving, TensorFlow Lite for mobile and embedded devices).
- **PyTorch**, developed by Facebook AI Research, uses a dynamic computation graph, which means operations are executed immediately. This “define-by-run” paradigm allows intuitive debugging and experimentation, making it the preferred choice in research labs.

When to choose which:

- TensorFlow is advantageous when transitioning from prototype to large-scale deployment due to its mature serving tools.
- PyTorch is generally favored in academic research and rapid prototyping because of its flexibility and Pythonic interface.

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

1. **Exploratory Data Analysis (EDA):** Jupyter Notebooks facilitate interactive exploration of datasets, allowing researchers to visualize distributions, detect anomalies, and iteratively refine preprocessing steps with immediate feedback through inline plots.
2. **Model Prototyping and Experiment Documentation:** Notebooks enable step-by-step development and testing of machine learning models. They also serve as a transparent record of the experiment, combining executable code with narrative explanations, equations (via LaTeX), and results, promoting reproducibility and collaborative research.

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

While Python’s native string methods suffice for rudimentary text manipulation (e.g., splitting or lowercasing), they lack the linguistic sophistication required for robust Natural Language Processing (NLP). spaCy provides:

- **Tokenization, Part-of-Speech tagging, syntactic parsing, and Named Entity Recognition (NER)** powered by pre-trained statistical models.
- **Pipeline components** that work efficiently at scale due to optimized Cython backends.
- The ability to customize and train domain-specific models with ease.

This comprehensive pipeline yields context-aware text analysis that goes far beyond literal string pattern matching.

2. Comparative Analysis

Aspect	Scikit-learn	TensorFlow
Target Applications	Primarily classical machine learning methods such as linear models, decision trees, clustering, and ensemble techniques. Well-suited for tabular data.	Primarily deep learning architectures, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformers for complex tasks such as image recognition and sequence modeling.
Ease of Use for Beginners	High; intuitive API, consistent estimators, and simple syntax make it ideal for students and practitioners learning ML fundamentals.	Moderate to steep; requires understanding of tensors, computational graphs, and training loops, though high-level APIs like Keras ease this barrier.
Community Support	Large, stable community; comprehensive documentation and numerous educational resources for ML tasks.	Extensive community and industrial support; abundant tutorials for cutting-edge deep learning applications and production deployment.

🎓 ▯ PART 3: Ethics & Optimization (10%)

Ethical Reflection:

- **Bias in MNIST:** Although MNIST is relatively neutral, variations in handwriting styles across demographics may cause biased misclassifications if the training set lacks diversity.
- **Bias in Amazon Reviews:** Rule-based sentiment may fail to detect sarcasm, idioms, or cultural nuances, leading to misinterpretation and potential bias against certain product types or customer groups.

Mitigation Strategies:

- Use tools such as **TensorFlow Fairness Indicators** to evaluate model fairness across demographic slices.

- Expand training data to include diverse and representative examples.
- Augment spaCy’s rule-based sentiment with supervised sentiment classifiers fine-tuned on domain-specific reviews to reduce misinterpretation.