

Part 1: Theoretical Understanding

Name: Brian Munene Gitau

1. Short Answer Questions

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

TensorFlow	PyTorch
1. Developed by Google, uses static computational graphs	Developed by Facebook, uses dynamic computational graphs
2. Strong production deployment capabilities with TensorFlow Serving, TensorFlow Lite	Preferred for research and prototyping due to easier debugging
3. Comprehensive ecosystem	More intuitive for Python developers
4. Better for large-scale production systems	Excellent for rapid experimentation

When to choose:

- TensorFlow: Production systems, mobile deployment, when using TensorFlow Extended (TFX)
- PyTorch: Research projects, rapid prototyping, when dynamic graphs are beneficial

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

- Exploratory Data Analysis and Visualization: Jupyter Notebooks provide an interactive environment for data scientists to explore datasets, create visualizations, and understand data distributions before model development.
- Model Prototyping and Experimentation: Researchers can quickly test different model architectures, hyperparameters, and preprocessing techniques while maintaining a record of all experiments with code, outputs, and explanations in a single document.

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

- Pre-trained models for multiple languages with accurate POS tagging, dependency parsing, and NER
- Efficient tokenization that understands linguistic rules rather than simple string splitting
- Semantic understanding through word vectors and similarity detection
- Rule-based matching with linguistic patterns rather than just regex
- Pipeline architecture for efficient processing of large text volumes

2. Comparative Analysis

Aspect	Scikit-learn	TensorFlow
1. Target Applications	Classical ML algorithms (SVM, decision trees, clustering)	Deep learning (neural networks, CNNs, RNNs)
2. Ease of Use for Beginners	Extremely easy with consistent API design	Steeper learning curve, more complex concepts
3. Community Support	Excellent documentation, many examples	Large community, extensive resources but more complex
4. Performance	Optimized for traditional ML on CPU	GPU acceleration, distributed training
5. Deployment	Simple models, REST APIs	Comprehensive deployment options