

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

- ✓ TensorFlow is a deep learning framework developed by Google. It offers both high-level APIs (like Keras) and low-level control for production-ready deployments. It is ideal for large-scale model deployment, especially with TensorFlow Serving and TensorFlow Lite.
- ✓ PyTorch, developed by Facebook, is known for its dynamic computational graph, making it more intuitive and flexible, especially during research and experimentation.

When to choose:

- ❖ Choose PyTorch for research, prototyping, and academic use due to its ease of debugging and dynamic computation.
- ❖ Choose TensorFlow for production environments, mobile deployment, and performance optimization with tools like TensorFlow Lite or TensorFlow Extended (TFX).

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

- ✓ Exploratory Data Analysis (EDA): Jupyter Notebooks allow data scientists to visualize and analyze datasets interactively using charts, plots, and markdown documentation in a single environment.
- ✓ Model Prototyping and Testing: They are useful for rapidly testing machine learning models, tweaking hyperparameters, and sharing experiments with visual outputs and code explanations.

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

spaCy provides pre-trained language models and advanced NLP pipelines that can perform:

- Tokenization, Named Entity Recognition (NER), Part-of-Speech (POS) tagging, and Dependency Parsing.
- These go far beyond basic string operations like `.split()` or `.find()` by understanding context, grammar, and semantic structure, thus enabling more accurate and scalable NLP applications.

2. Comparative Analysis: Scikit-learn vs. TensorFlow

Criteria	Scikit-learn	TensorFlow
Target Applications	Classical machine learning (e.g., decision trees, SVMs, clustering, regression)	Deep learning and neural networks (e.g., CNNs, RNNs, transformers)
Ease of Use for Beginners	Very beginner-friendly with consistent APIs and simple model fitting (fit, predict)	More complex, especially at the low-level API; better for those familiar with neural networks
Community Support	Strong support in academic and applied ML communities; excellent documentation.	Very large community with extensive resources, especially for deep learning and production deployment.

AI Project Summary Report

- Scikit-learn: Strong academic and developer support
- TensorFlow: Extensive community and deployment ecosystem

3. Ethical Reflection

The development and application of Artificial Intelligence (AI) raise several ethical considerations that must be addressed to ensure responsible innovation. One major concern is data privacy - models trained on sensitive or personal data must ensure compliance with legal standards like GDPR and prioritize anonymization and informed consent.

Additionally, algorithmic bias is a persistent issue. AI systems may perpetuate social inequalities if the training data reflect historical or societal biases. This can have serious consequences in areas such as hiring, lending, or law enforcement. Developers must actively audit models and datasets for fairness and inclusivity.

Finally, transparency and accountability are essential. As AI becomes more integrated into critical systems, stakeholders must understand how decisions are made and who is responsible for them. Creating explainable AI (XAI) and maintaining open communication with users is vital.

Ethical AI development is not just a technical challenge but a moral responsibility to society.