---

## 🧠 AI Tools & Frameworks — Short Answers and Comparative Analysis

---

### 1. Short Answer Questions

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

**TensorFlow:** Uses static computation graphs (define-before-run), great for large-scale deployment and production.

**PyTorch:** Uses dynamic computation graphs (define-by-run), better for flexibility and research.

✅ Choose TensorFlow for production and scalability.

✅ Choose PyTorch for experimentation and academic research.

---

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

- **Model Prototyping:** Quickly test AI models and visualize outputs in real time.
- **Data Analysis & Visualization:** Explore datasets using libraries like Pandas, Matplotlib, and Seaborn interactively.

---

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

spaCy provides **advanced NLP features such as tokenization, part-of-speech tagging, and named entity recognition,** which basic Python string functions can't perform. It processes text linguistically rather than just manipulating raw strings.

---

### 2. Comparative Analysis: Scikit-learn vs TensorFlow

Comparative Analysis Compare Scikit-learn and TensorFlow in terms of:

- Target applications (e.g., classical ML vs. deep learning).
- Ease of use for beginners.
- Community support.

| Feature | Scikit-learn | TensorFlow |
|---|---|---|
| **Target Applications** | Classical ML (e.g., regression, SVMs, clustering) | Deep learning & neural networks |
| **Ease of Use** | Easier for beginners, simpler API | Steeper learning curve, more complex syntax |
| **Community Support** | Large, stable, and mature | Very large, fast-evolving with Google support |