AI Tools and Applications - Part 1: Theoretical Understanding

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

TensorFlow and PyTorch are both open-source machine learning frameworks widely used in AI development, but they differ in how they operate and who they serve best.

TensorFlow:
- Uses static computation graphs (define-and-run)
- Steeper learning curve, more complex for beginners
- Strong deployment support (e.g., TensorFlow Lite, TF.js)
- Backed by Google, optimized for production

PyTorch:
- Uses dynamic computation graphs (define-by-run)
- Easier to use and debug (Pythonic syntax)
- Rapid prototyping and research-friendly
- Backed by Facebook, gaining popularity

When to Choose:
- Choose TensorFlow for production, mobile, or web deployment.
- Choose PyTorch for experimentation, flexibility, and academic use.

2. Describe two use cases for Jupyter Notebooks in AI development.

1. Exploratory Data Analysis (EDA):
Jupyter makes it easy to load, inspect, and visualize datasets interactively. AI developers can explore trends and prepare data before modeling.

2. Model Development:
Developers can train and test machine learning models step-by-step, visualize results (like accuracy plots), and easily iterate without restarting scripts.

3. How does spaCy enhance NLP tasks compared to basic Python string operations?

spaCy is a powerful NLP library that offers advanced language features which basic Python string operations cannot perform.

Compared to basic Python:

- spaCy provides accurate tokenization, POS tagging, and named entity recognition.
- It can identify names, places, and organizations from raw text.
- Offers dependency parsing and text embeddings for deeper analysis.
- Faster, optimized, and production-ready for large-scale NLP applications.

Conclusion:

spaCy simplifies and improves NLP by offering advanced text processing features out-of-the-box, unlike basic string tools in Python.