SHORT ANSWER QUESTIONS

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

Difference: PyTorch uses a “dynamic graph,” meaning you can change the code structure while it runs, making it easier to debug. TensorFlow (historically) uses a “static graph,” where you define the structure first and run it later, which is better for optimization.

Selection: I would choose PyTorch for research and learning because it is more flexible and “Pythonic.” I would choose TensorFlow for deploying finished products to mobile or web, as it has robust deployment tools (TFX, TF Lite).

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

Exploratory Data Analysis (EDA): It allows developers to run code in small chunks to inspect data and generate charts instantly without re-running the entire script.

Documentation and Teaching: It combines live code with rich text explanations, making it perfect for creating tutorials or sharing step-by-step research findings.

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

Basic Python only treats text as a sequence of characters (e.g., splitting a sentence by spaces). spaCy understands the linguistic structure of the text. It can identify parts of speech (nouns vs. Verbs), finding named entities (like “Lagos” or “Google”), and understanding word dependencies, which is impossible with simple string methods.

COMPARATIVE ANALYS SCIKIT-LEARN VS. TENSORFLOW

Scikit-learn and TensorFlow serve distinct roles in the machine learning landscape. **In terms of target applications**, Scikit-learn is best suited for classical machine learning tasks using structured data (like tables and CSVs), such as regression and clustering. In contrast, TensorFlow is optimized for deep learning, making it the better choice for unstructured data like images, audio, and complex neural networks.

Regarding ease of use, Scikit-learn is very high; it offers a simple, consistent interface (using standard commands like .fit() and .predict()) that is easy for beginners to master. TensorFlow has a moderate to low ease of use with a steeper learning curve, as it requires understanding complex underlying concepts like tensors and computational graphs.

Finally, considering community support, Scikit-learn provides a stable and mature environment with excellent documentation focused on standard algorithms and analysis. TensorFlow benefits from massive corporate backing by Google, resulting in a huge ecosystem of tutorials, pre-trained models, and deployment tools