Theoretical Understanding

1. Short Answer Questions

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

Feature / Aspect	TensorFlow	PyTorch
How it works	Builds a **computation graph** behind the scenes before running the code.	Runs **line by line**, just like normal Python.
Ease of use	Slightly harder to learn at first because of graphs and sessions.	Easier for beginners you can see results as you go.
Debugging	Harder to debug because you don't see live outputs until the graph runs.	Easier to debug errors appear immediately.
Best for	**Production and deployment** (apps, mobile, web).	**Research and learning** (trying out new ideas).
Speed and performance	Very fast when deployed; works well on servers and devices.	Fast too, but mainly used for experiments and model testing.
Tools and support	Works well with **TensorFlow Lite** and **TensorFlow Serving** for scaling.	Very popular in universities and research labs.
Summary	Great for **scaling and deploying** AI models.	Great for **building and learning** new models.

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

Aspect	**Description**
Testing and Learning	Jupyter lets you try small pieces of code, see results right away, and fix mistakes step by step. AI developers use it to test ideas quickly, like training a small model or checking data.
Sharing Results	You can combine code, notes, and graphs in one notebook, making it easy to show your work clearly to teachers, teammates, or managers.
Overall Use	Jupyter is great for both **experimenting** and **explaining** in AI, it helps you learn and communicate results visually.

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

Basic Python string tools (`split`, `find`, etc.) only look at raw text they don't understand language.
spaCy goes deeper: it breaks text into **tokens**, finds **parts of speech**, and can detect **names, places, and brands** using its built-in models.

Instead of just matching words, spaCy understands **meaning and context,** which makes it much better for real NLP work like chatbots or sentiment analysis.

Python sees text as characters; **spaCy sees text as language.**

2. Comparative Analysis

- Compare Scikit-learn and TensorFlow in terms of:

Aspect	**Scikit-learn**	**TensorFlow**

Target Applications	Used for **classical machine learning** tasks such as regression, decision trees, and clustering.	Used for **deep learning** tasks like image recognition, speech, and large neural networks.
Example Use Case	Predicting simple things like house prices or flower types (e.g., Iris dataset).	Training models on complex data like images, videos, or big datasets.
Ease of Use (Beginners)	Easy to start one can train and test models with simple commands like `model.fit()` and `model.predict()`.	Needs more setup defining layers, optimizers, and tensors, but gives more control for big projects.
Best For	Learning basics and small projects.	Advanced deep learning and real-world production.
Community Support	Strong academic and beginner community with many simple tutorials.	Large global community backed by Google, focused on scaling and deployment.
Summary	Great for quick, simple models.	Great for complex neural networks and production-level AI.