AI Tools Assignment – Part 1 (Theoretical Understanding)

Q1: Differences Between TensorFlow and PyTorch

TensorFlow uses static computation graphs (with modern eager execution), strong deployment tools, and is industry■focused. PyTorch uses dynamic graphs, is easier to debug, and is preferred in research.

When to choose TensorFlow:

• Production deployment

• Mobile/web via TFLite or TF.js

• TensorBoard + TFX pipelines

When to choose PyTorch:

• Rapid prototyping

• Research and experimentation

• Pythonic workflows

Q2: Two Use Cases for Jupyter Notebooks

1. Interactive experimentation: test models, tweak parameters, view immediate outputs.

2. Data visualization & documentation: mix code, charts, and explanations for reproducibility.

Q3: How spaCy Enhances NLP

• Linguistically aware tokenization

• Named Entity Recognition (NER)

• Pretrained efficient pipelines

• More accurate NLP than basic Python string operations

Comparative Analysis: Scikit■learn vs TensorFlow

Target Applications:

• Scikit■learn → classical ML (SVM, trees, regression)

• TensorFlow → deep learning (CNNs, RNNs, transformers)

Ease of Use:

• Scikit■learn → very beginner■friendly (fit/predict)

• TensorFlow → more complex (architectures, training loops)

Community Support:

• Scikit■learn → strong academic & classical ML support

• TensorFlow → massive industry community & deployment ecosystem