AI/ML Assignment - Part 1: Theoretical Understanding

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, making it optimized for deployment, especially in production environments. PyTorch uses dynamic computation graphs, allowing for more flexibility and easier debugging.

Choose TensorFlow when you need scalable deployment (e.g., mobile, web) and production-ready tools.

Choose PyTorch for research, rapid prototyping, and when you want a more Pythonic interface.

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

1. Exploratory Data Analysis (EDA): Enables interactive visualization and preprocessing of datasets.
2. Model Prototyping: Allows step-by-step development, debugging, and testing of machine learning models.

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

spaCy provides efficient and accurate NLP pipelines with pre-trained models for tokenization, POS tagging, and NER. It handles linguistic nuances better than basic string operations and can identify entities like 'New York' as a single geopolitical entity (GPE).

2. Comparative Analysis

Compare Scikit-learn and TensorFlow:

Target Applications:

- Scikit-learn is ideal for classical machine learning models like SVMs, decision trees, and ensemble methods.

- TensorFlow is designed for deep learning models including CNNs, RNNs, and large-scale neural networks.

Ease of Use for Beginners:

- Scikit-learn has a simple, consistent API and is easier for beginners to learn and use.
- TensorFlow can be more complex due to its architecture and deep learning focus.

Community Support:

- Scikit-learn has strong academic and research community support.
- TensorFlow has robust industrial support, especially from Google, and extensive tools for production.