AI Tools Assignment - Theoretical Section

1. Compare TensorFlow and PyTorch

TensorFlow is a comprehensive, production-ready deep learning framework developed by Google. It is known for its robust deployment options (TensorFlow Lite, TensorFlow.js) and scalability. PyTorch, developed by Facebook, is more flexible, intuitive, and widely used in academic research.

Key differences:

- TensorFlow uses static graphs (TF 1.x) but now supports eager execution in TF 2.x.
- PyTorch uses dynamic computation graphs by default, making debugging easier.
- TensorFlow is more production-oriented, while PyTorch is more research-oriented.

2. Use Cases of Jupyter Notebooks in AI Development

Jupyter Notebooks are essential tools in AI development. They support:

- Step-by-step code execution, which aids debugging and learning.
- Easy integration of code, visualizations, and documentation.
- Ideal for exploratory data analysis (EDA), model building, and presentation.

3. spaCy vs Python String Operations

spaCy is an NLP library built for production use and provides high-level natural language processing features such as tokenization, part-of-speech tagging, and named entity recognition (NER). Python string operations like split(), replace(), and find() are low-level and cannot interpret language context or grammar.

Example: spaCy can detect "Apple" as an organization in "Apple is buying a startup", while basic string functions cannot.

4. Scikit-learn vs TensorFlow

Scikit-learn is a Python library for classical machine learning (SVM, kNN, decision trees), suitable for small to medium datasets. TensorFlow is a deep learning framework ideal for large-scale neural networks (CNNs, RNNs) and deployment.

Comparison:

- Scikit-learn is beginner-friendly and easy to implement.
- TensorFlow is better for complex tasks like image recognition, NLP, and scalable models.

5. Named Entity Recognition (NER) in spaCy

spaCy uses pre-trained statistical models to recognize named entities in text. NER identifies entities such as people, organizations, dates, and locations.

Example:

Text: "Google was founded in 1998."

Entities:

- Google: ORG
- 1998: DATE

6. When to Use Deep Learning over Classical ML

Use deep learning when:

- Data is large and complex (images, audio, text).
- Problem involves complex patterns (e.g., image classification, speech recognition).
- You need transfer learning or pre-trained models (e.g., BERT, GPT).

Use classical ML for:

- Tabular data and small datasets.
- Quick, interpretable models (e.g., linear regression, decision trees).