1. Short Answer Questions

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

TensorFlow, developed by **Google**, is widely used in **production environments**, whereas PyTorch, created by **Meta (Facebook)**, is more popular in **research and experimentation**.

While TensorFlow traditionally relied on **static computation graphs**, making it optimized for large-scale applications, PyTorch uses **dynamic computation graphs**, allowing for easier debugging and rapid prototyping.

TensorFlow excels in **mobile AI applications** (e.g., TensorFlow Lite) and cloud-based deployments, while PyTorch is preferred for **academic research** and projects requiring **custom model development**.

When choosing between the two, TensorFlow is ideal for **enterprise-level AI solutions**, whereas PyTorch is better suited for **exploratory work and fast iterations**.

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

Jupyter Notebooks are widely used for **interactive data exploration**, allowing AI developers to visualize datasets in real time using tools like **Matplotlib, Seaborn, and Pandas**. This makes it easier to perform **Exploratory Data Analysis (EDA)** before training machine learning models.

It is used in **reproducible AI experiments**, where researchers and engineers can execute code step by step, tweak parameters, and document findings within the same notebook. This structured approach is especially useful for **collaborative research**, enabling teams to share results with annotations and visualizations.

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

spaCy provides **pre-trained models** for tasks like **tokenization, named entity recognition (NER), and dependency parsing**, making it far more efficient than basic Python string operations. While Python's `.split()` method simply separates text based on spaces, spaCy's tokenizer understands **punctuation, contractions, and multi-word expressions**, ensuring more accurate text processing.

Additionally, spaCy offers **Named Entity Recognition (NER)**, which can identify **people, places, organizations**, and other entities, whereas basic string operations lack semantic understanding. This makes spaCy ideal for **advanced NLP applications** like chatbots, document analysis, and automated text classification.

2. Comparative Analysis

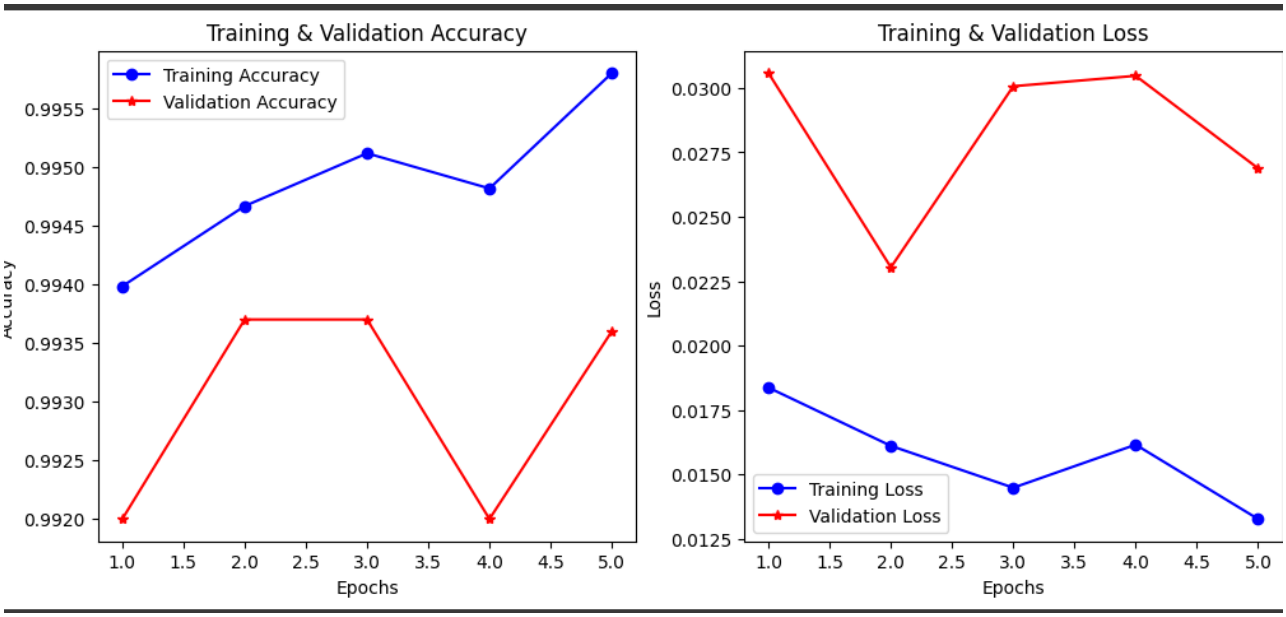
- Compare Scikit-learn and TensorFlow in terms of:
    - Target applications (e.g., classical ML vs. deep learning).
    - Ease of use for beginners.
    - Community support.

Scikit-learn is primarily designed for **classical machine learning tasks** such as regression, classification, and clustering, while TensorFlow is optimized for **deep learning applications**, including neural networks and large-scale AI models.

For beginners, Scikit-learn offers a **simpler and more intuitive API**, making it easier to implement machine learning models with minimal code, whereas TensorFlow requires a deeper understanding of **tensors, computational graphs, and neural network architectures**, making it more complex for newcomers.

In terms of community support, both libraries have **strong developer communities**, but TensorFlow benefits from **Google's backing**, extensive documentation, and large-scale industry adoption, while Scikit-learn is widely used in **academic research and traditional ML projects**.

SCREENSHOT OF OUTPUTS



Ethical Considerations in AI Models

AI models, including those trained on datasets like MNIST for digit classification or Amazon Reviews for sentiment analysis, can unintentionally inherit biases from their training data. These biases can affect the fairness and accuracy of predictions, leading to skewed results that disproportionately favor certain groups or patterns.

For example, in the MNIST dataset, if certain digits (like "1" or "7") appear more frequently than others, the model might become better at recognizing those digits while struggling with less common ones. Similarly, in Amazon Reviews sentiment analysis, biases can emerge if the dataset contains more reviews from a specific demographic or product category, leading to unfair sentiment predictions.

To address these concerns, tools like TensorFlow Fairness Indicators can help analyze model performance across different subgroups, ensuring that predictions remain balanced and unbiased. Additionally, spaCy's rule-based systems can refine text processing by filtering out biased language patterns, improving the fairness of NLP models.

Ultimately, ethical AI development requires continuous monitoring, diverse training data, and fairness-aware evaluation techniques to ensure that models make equitable and reliable predictions.