Part 1: Theoretical Understanding (40%)

Question 1

Here’s a detailed summary of the primary differences between TensorFlow and PyTorch, two prominent machine learning frameworks. TensorFlow, developed by Google Brain in 2015, uses static computation graphs which follow a define-then-run approach, making it more complex and less intuitive compared to its counterpart. It excels in production and deployment, boasting tools like TensorFlow Serving and TensorFlow Lite. Its ecosystem is extensive, incorporating Keras and TensorBoard, and it is optimized for large-scale production with strong documentation support. Conversely, PyTorch, created by Facebook’s AI Research Lab in 2016, employs dynamic computation graphs, enabling a more Pythonic and flexible coding experience, particularly beneficial for research and experimentation. It is simpler for debugging and facilitates custom model architectures. While TensorFlow is preferable for production and scalability, PyTorch is favored for research and quick prototyping due to its ease of use and integrated environment with Python's data science tools.

Question 2

Jupyter Notebooks serve two significant purposes in AI development: Exploratory Data Analysis (EDA) and Model Prototyping. For EDA, they facilitate interactive analysis and visualization of datasets by allowing data scientists to clean data, generate statistics, and create visualizations using libraries such as Pandas, Matplotlib, and Seaborn. This process aids in understanding data patterns and preparing datasets for AI model training. Additionally, Jupyter Notebooks are used for rapid model prototyping and experimentation, enabling AI developers to build, test, and compare various machine learning or deep learning models efficiently. Their interactive environment supports step-by-step execution and smooth integration with frameworks like TensorFlow, PyTorch, and scikit-learn.

Question 3

spaCy enhances Python's basic string operations by offering advanced Natural Language Processing (NLP) features capable of understanding the linguistic structure and meaning of text. It performs essential tasks such as tokenization, lemmatization, part-of-speech tagging, named entity recognition (NER), dependency parsing, and sentence segmentation, significantly surpassing the capabilities of basic string functions. For example, spaCy can identify the relationship between “running” and “run” as different forms of the same verb and distinguish that “Apple” refers to a company rather than a fruit. Furthermore, spaCy is designed for high speed and accuracy, utilizes pre-trained language models, and seamlessly integrates with machine learning pipelines, making it a more powerful and efficient choice for complex NLP tasks compared to standard Python string manipulations.

2. Comparative Analysis

Scikit-learn and TensorFlow cater to different aspects of machine learning. Scikit-learn is designed for classical machine learning applications, including regression, classification, and clustering, particularly suitable for smaller, structured datasets. It is user-friendly due to its straightforward API, making it ideal for beginners and quick testing. Conversely, TensorFlow specializes in deep learning and neural networks, allowing for the construction of sophisticated AI models applicable in areas such as image recognition, speech processing, and natural language comprehension. While TensorFlow has a more challenging learning curve, frameworks like Keras enhance its accessibility. Both libraries boast substantial community support: Scikit-learn is favored in academic and research settings, whereas TensorFlow enjoys robust industry support from Google and is widely employed for production AI systems.