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?

The primary difference lies in their approach to defining the computation graph: **TensorFlow** typically uses a **Static Graph**, where the entire model structure is defined and compiled before execution, making it highly efficient for deployment and large-scale serving. **PyTorch** uses a **Dynamic Graph** (an imperative style), which is built on the fly as the code executes. This dynamic nature allows for easier debugging and more flexible model architectures, which is why it is often favored in academic research.

- **Choose TensorFlow** when the main goal is **production deployment**, especially for deploying models across diverse platforms like mobile, edge devices, or large cloud systems (using tools like TensorFlow Serving).
-
- **Choose PyTorch** when the main goal is **rapid prototyping, research, or complex model experimentation**, as its dynamic nature is more intuitive and flexible for researchers.
-

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

1st **Exploratory Data Analysis (EDA) and Visualization:** Jupyter Notebooks (and Google Colab) are ideal for EDA because they support **interactive, cell-by-cell execution**. Developers can load and clean data, perform statistical analysis, and immediately visualize results (charts, histograms) next to the code that produced them, allowing for a deep, visual understanding of the data before model training begins.

2nd

3rd **Model Prototyping and Incremental Development:** The notebook environment excels at **rapidly building and testing model components**. A developer can define the data loading in one cell, the model architecture in the next, and the training loop in another. This allows for quick iteration on model architecture and hyperparameter tuning without needing to recompile or rerun an entire script.

4th

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

spaCy provides sophisticated, linguistic understanding that goes far beyond simple text manipulation available in basic Python string operations (like .split() or .replace()):

- **Linguistic Depth:** Basic string operations only break text by simple separators (like spaces). spaCy provides highly accurate **Tokenization** and performs **Part-of-Speech (POS) tagging** and **Dependency Parsing**, identifying the grammatical role and relationships between words.
-

- **Entity Recognition:** spaCy automatically performs **Named Entity Recognition (NER)** (as used in Task 3) to identify and classify real-world entities like people, organizations, locations, and products, a crucial and impossible task for basic string methods.
-
- **Efficiency:** spaCy is optimized for **speed and scalability** in production environments, making it far more efficient for processing large volumes of text data compared to writing custom, slow text processing loops in pure Python.
-

---

2. Comparative Analysis

Compare Scikit-learn and TensorFlow in terms of:

Target Applications (e.g., classical ML vs. deep learning)

- **Scikit-learn** is the standard toolkit for **Classical Machine Learning**. It is best suited for traditional algorithms like Decision Trees (Task 1), Support Vector Machines, Linear Regression, and clustering algorithms, which typically work well with structured, medium-sized datasets.
-
- **TensorFlow** is the leading framework for **Deep Learning**. It is essential for complex architectures like CNNs (Task 2), Recurrent Neural Networks (RNNs), and Transformers, designed to handle massive, unstructured data (images, text, audio) and complex, multi-layered models.
-

Ease of Use for Beginners

- **Scikit-learn** has an **extremely intuitive and simple API**. Its standardized fit(), predict(), and transform() methods make it easy for beginners to quickly build and evaluate models without needing a deep understanding of the underlying mathematical graph operations.
-
- **TensorFlow** has a **steeper learning curve**. Beginners must grasp concepts like tensors, computational graphs, specific layer types, and managing GPU resources, making the initial setup more complex.
-

Community Support

- **Scikit-learn** has extensive documentation and strong community support, particularly focused on **academic and applied statistics**.
-
- **TensorFlow** has massive, global community support driven by its adoption in both **academia and major industry applications**. It offers comprehensive supplementary tools (like TensorBoard) that reflect its wide industry use.
-
-

Part 2 task 3 and task 2 visualisation respectively;

```
--- Review 1 ---
Text: This tea is very flavorful, and very relaxing.  Best part is it contains no caffeine so I ...
  Extracted Products: None detected
  Extracted Brands/ORGs: None detected
  Sentiment: Positive (Score: 0.7764)
------------------------------------------------------------------------
--- Review 2 ---
Text: Nescafe Taster's Choice Instant Coffee is as good if not better than the local coffee shop...
  Extracted Products: None detected
  Extracted Brands/ORGs: ["Nescafe Taster's Choice Instant Coffee", "Nescafe Taster's Choise"]
  Sentiment: Positive (Score: 0.1265)
------------------------------------------------------------------------
--- Review 3 ---
Text: <a href="http://www.amazon.com/gp/product/B000ENWSBW">Larabar Fruit & Nut Food Bar, Cherry...
  Extracted Products: None detected
  Extracted Brands/ORGs: ['Fruit & Nut Food Bar', 'Apple']
  Sentiment: Positive (Score: 0.9286)
------------------------------------------------------------------------
--- Review 4 ---
Text: I was expecting more (much more actually) after reading the reviews.  And for the record, ...
  Extracted Products: None detected
  Extracted Brands/ORGs: ['Pumpkin Spice', 'Smell']
  Sentiment: Positive (Score: 0.9910)
------------------------------------------------------------------------
--- Review 5 ---
Text: Good stuff tastes nice like it after dinner before bed nice price, Good stuff tastes nice ...
```

```
--- Visualization of 5 Sample Predictions ---
```