AI/ML Frameworks and Tools: Quick Reference Summary

---

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 computational graph:

- **PyTorch** uses a **Dynamic Computational Graph** (or "define-by-run"), meaning the graph is built on the fly as the code executes.1 This makes it feel more **"Pythonic,"** easier to debug, and highly flexible for rapid prototyping and research.

- **TensorFlow** (especially TensorFlow 1.x) traditionally used a **Static Computational Graph** (or "define-then-run"), requiring the entire model architecture to be defined before execution.2 **TensorFlow 2.x**, however, adopted an "eager execution" default, which resembles PyTorch's dynamic approach, making it much more user-friendly.3

Choosing a Framework:

Framework	Best Choice When...	Rationale
PyTorch	**Rapid research, prototyping, and academic projects.**	Its dynamic graph makes debugging easier, and its Pythonic nature simplifies experimentation with complex, custom model architectures.
TensorFlow	**Large-scale production deployment, mobile, and web applications.**	It has a more mature ecosystem for deployment (e.g., TensorFlow Serving, TensorFlow Lite) and strong support for distributed training and specialized hardware (like TPUs).

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

Jupyter Notebooks are an interactive environment that combines code, output, visualizations, and documentation in a single document, making them ideal for

- **Exploratory Data Analysis (EDA) and Visualization:**

  AI development begins with understanding the data. Jupyter Notebooks allow developers to load datasets, clean and preprocess data step-by-step, and immediately **visualize distributions, correlations, and anomalies** using libraries like Matplotlib and Seaborn, all while documenting the process with Markdown cells.5 This interactive feedback loop accelerates data preparation.

- **Rapid Model Prototyping and Iteration:**

  Notebooks support a cell-by-cell execution workflow, which is perfect for building and testing small segments of a model. Developers can **define a model architecture**, **train it on a subset of data**, and **evaluate its performance** with visualizations, all within a few cells. If a tweak is needed (e.g., changing a learning rate or activation function), only the relevant cells need to be re-run, enabling fast iteration and hypothesis testing.

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

spaCy provides a **high-level, efficient, and production-ready** framework for Natural Language Processing (NLP), delivering sophisticated linguistic understanding that basic Python string operations (like .split() or regular expressions) cannot easily match.6

Feature	spaCy Enhancement	Limitation of Basic Python String Operations
Tokenization	Provides **intelligent tokenization** (splitting text into meaningful units like words and punctuation) that handles contractions (e.g., "don't" → "do", "n't") and special characters correctly.	Basic string methods or simple regex often fail with edge cases, leading to inaccurate token counts and messy preprocessing.
Linguistic Annotation	Automatically adds **linguistic metadata** like **Part-of-Speech (POS) tagging** (e.g., noun,	This level of linguistic analysis is impossible without building and

Feature	spaCy Enhancement	Limitation of Basic Python String Operations
	verb), **Dependency Parsing** (grammatical relations), and **Named Entity Recognition (NER)** (identifying people, locations, dates).	training a complex statistical model from scratch. String operations only deal with characters and patterns.
Normalization	Efficiently performs tasks like **lemmatization** (reducing words to their dictionary form, e.g., "running" → "run") to ensure different word forms are treated as the same concept.	Basic operations can only perform crude normalization like lowercasing. Lemmatization requires knowledge of morphology, which standard string operations lack.

---

2. Comparative Analysis: Scikit-learn vs. TensorFlow

Aspect	Scikit-learn	TensorFlow
Target Applications	Primarily **Classical Machine Learning** on **structured/tabular data**. Includes algorithms for classification (e.g., SVM, Random Forest), regression (e.g., Linear Regression), clustering (e.g., K-Means), and dimensionality reduction.	**Deep Learning** on **unstructured data** (images, text, audio). Primarily used for building complex neural networks (NNs), Computer Vision (CNNs), and Natural Language Processing (RNNs, Transformers).
Ease of Use for Beginners	**Extremely high.** It features a consistent, intuitive API (.fit(), .predict()) and is designed for simple, fast implementation of traditional algorithms with minimal setup.	**Moderate to high (with Keras).** While TensorFlow 2.x with the Keras API is much easier, building and debugging complex custom neural networks still requires a deeper understanding of underlying concepts like tensors, graphs, and hardware acceleration.

Aspect	Scikit-learn	TensorFlow
Community Support	**Very large and stable.** It's the de facto standard for traditional ML, with extensive documentation, tutorials, and widespread adoption in both academia and industry for non-deep learning tasks.	**Massive and industry-leading.** Backed by Google, it has an immense, active community and a full-fledged ecosystem (TensorBoard, TF Serving, TF Lite) for research, deployment, and cross-platform use.

Screenshots of model outputs

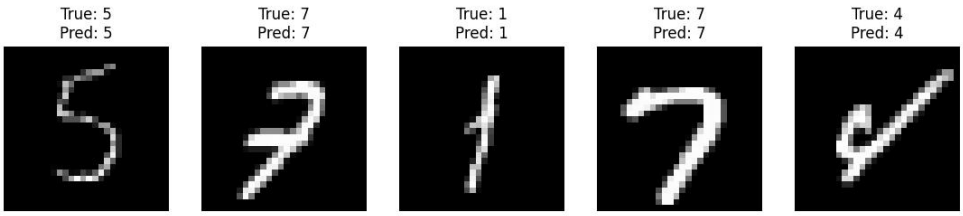
TASK 1 :

```
--- Starting Task 1: Scikit-learn (Iris) ---
Initial dataset shape: (150, 5)
Missing values check: No missing values in the Iris dataset.
Encoded classes: [0 1 2]
Training set size: 105 samples
Testing set size: 45 samples

Decision Tree Classifier Trained successfully.

--- Model Evaluation Metrics ---
Accuracy: 1.0000
Precision (Macro): 1.0000
Recall (Macro): 1.0000
--- Task 1 Complete ---
```

TASK2 :

```
--- Model Evaluation ---
Test Loss: 0.0218
Test Accuracy: 0.9929 (Goal: >0.95)
SUCCESS: Achieved greater than 95% test accuracy!
```

CNN Predictions on Sample Images



TASK 3 :

```
--- Processing First 5 Extracted Reviews ---

Review 1 Text: "Great CD: My lovely Pat has one of the GREAT voices of her generation. I have listened to this CD fo..."
  Extracted Entities (NER): 'Pat' (PERSON), 'STUUNNING' (ORG)
  Sentiment (Rule-Based): Positive

Review 2 Text: "One of the best game music soundtracks - for a game I didn't really play: Despite the fact that I ha..."
  Extracted Entities (NER): 'Chrono Trigger' (PRODUCT)
  Sentiment (Rule-Based): Positive

Review 3 Text: "Batteries died within a year ...: I bought this charger in Jul 2003 and it worked OK for a while. Th..."
  Extracted Entities (NER): None Detected
  Sentiment (Rule-Based): Negative

Review 4 Text: "works fine, but Maha Energy is better: Check out Maha Energy's website. Their Powerex MH-C204F charg..."
  Extracted Entities (NER): 'Maha Energy' (ORG), 'Maha Energy's' (ORG), 'Powerex MH-C204F' (ORG)
  Sentiment (Rule-Based): Neutral/Mixed

Review 5 Text: "Great for the non-audiophile: Reviewed quite a bit of the combo players and was hesitant due to unfa..."
  Extracted Entities (NER): 'VHS' (ORG)
  Sentiment (Rule-Based): Positive

--- Task 3 Complete ---
```