AI Tools and Applications: Mastering the AI Toolkit

Author: Ronald Zani

Date: November 2025

Part 1: Theoretical Understanding

1. Short Answer Questions

Q1: TensorFlow vs. PyTorch

TensorFlow (TF) uses a Static Graph (defined before runtime), making it better for production deployment (TensorFlow Serving, TFLite).

PyTorch (PT) uses a Dynamic Graph (Define-by-Run), making it generally easier to debug and ideal for research due to its flexibility.

Q2: Jupyter Notebook Use Cases

1. Exploratory Data Analysis (EDA): Allows incremental loading, cleaning, and visualization of data within the same document.

2. Rapid Prototyping: Enables quick iteration of model architectures and hyperparameter tuning by executing code blocks sequentially.

Q3: spaCy vs. Python Strings

spaCy provides linguistic awareness through pre-trained models, enabling tasks like Named Entity Recognition (NER), Part-of-Speech (POS) Tagging, and Dependency Parsing.

Basic string operations only handle characters, lacking any understanding of context or grammar.

2. Comparative Analysis: Scikit-learn vs. TensorFlow

Feature	Scikit-learn	TensorFlow (Keras)
Target Applications	Classical/Shallow ML on structured data	Deep Learning on unstructured data
Ease of Use for Beginners	Extremely High	Moderate
Community Support	Excellent, mature, stable	Vast, highly active

Part 2: Practical Implementation Outputs (Screenshots)

Task 1: Classical ML with Scikit-learn (Decision Tree)

The Decision Tree model achieved high performance on the Iris dataset, validating the classical ML workflow.

Evaluation Metrics:

Accuracy: 0.9333

Precision (Macro): 0.9444

Recall (Macro): 0.9333

Task 2: Deep Learning with TensorFlow (MNIST CNN)

Model Performance:

Test Accuracy: 0.9879 (Goal: >0.95)

Task 3: NLP with spaCy (NER and Sentiment)

NER and Sentiment Output:

Review 1: Entity - Sony (ORG)

Review 2: None found

Review 3: Entity - Acer (ORG)

Sentiment Results:

Review 1: Neutral

Review 2: Positive

Review 3: Negative

Part 3: Ethics & Optimization

1. Ethical Considerations and Mitigation

Model: MNIST (CNN)

Potential Bias Source: Data Collection Bias

Mitigation Strategy: Use TensorFlow Fairness Indicators (TFFI)

Model: Amazon Reviews

Potential Bias Source: Lexical Bias

Mitigation Strategy: Use POS Tagging & Dependency Parsing

2. Troubleshooting Challenge (Debugging TensorFlow)

Issue: Using sigmoid activation for multi-class classification

Correct Fix: Dense(10, activation='softmax') with sparse_categorical_crossentropy