Part 1: Theoretical Understanding

Q1: TensorFlow vs PyTorch

TensorFlow

- Developed by Google, uses computational graphs (static in TF1, eager execution in TF2).
- Ideal for production, strong integration with TensorFlow Serving, TensorBoard, and TFLite.

PyTorch

- Developed by Meta (Facebook).
- Dynamic computation graph (eager mode by default) more intuitive for research and experimentation.

When to choose:

- **PyTorch** → when rapid prototyping or academic work.
- **TensorFlow** → when deploying models at scale (especially on mobile or web).

Q2: Jupyter Notebook Use Cases

1. **Interactive experimentation:** Ideal for trying models, tuning parameters, and visualizing outputs instantly.
2. **Documentation & sharing:** Combines code, plots, and markdown in one file which are perfect for AI education and research reproducibility.

Q3: spaCy vs Basic Python String Ops

- Python's split() or regex can handle raw text but lack linguistic structure.
- **spaCy** provides tokenization, POS tagging, NER, and lemmatization using pre-trained language models, enabling context-aware text processing rather than pattern matching.

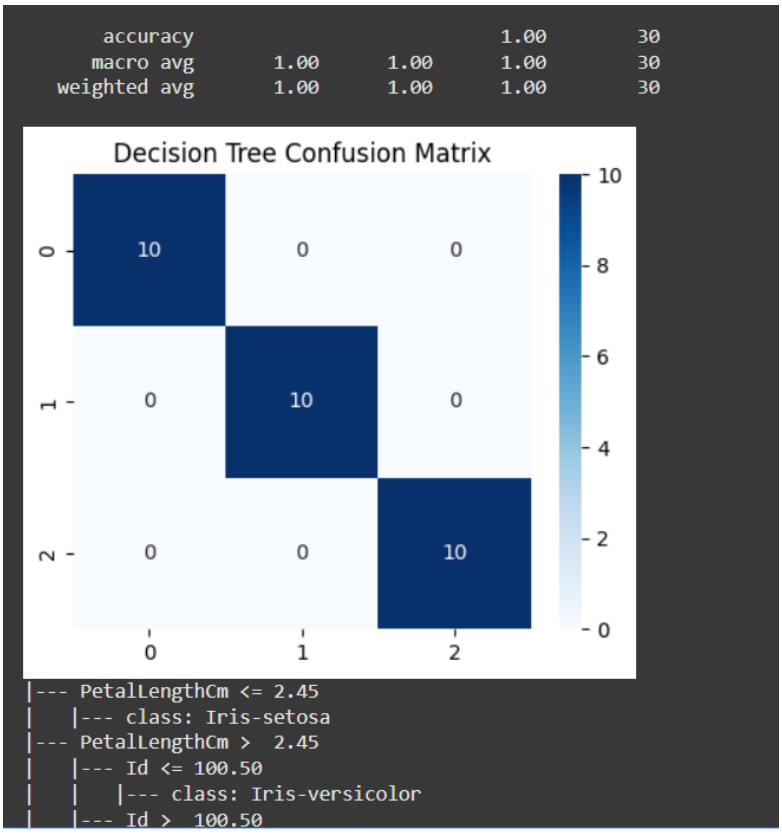
Comparative Analysis: Scikit-learn vs TensorFlow

Criteria	Scikit-learn	TensorFlow
Target Application	Classical ML (SVM, Decision Trees, etc.)	Deep Learning (CNNs, RNNs, Transformers)
Ease of Use	Simpler, great for beginners	More complex setup
Community Support	Mature & stable	Rapidly evolving, strong Google ecosystem

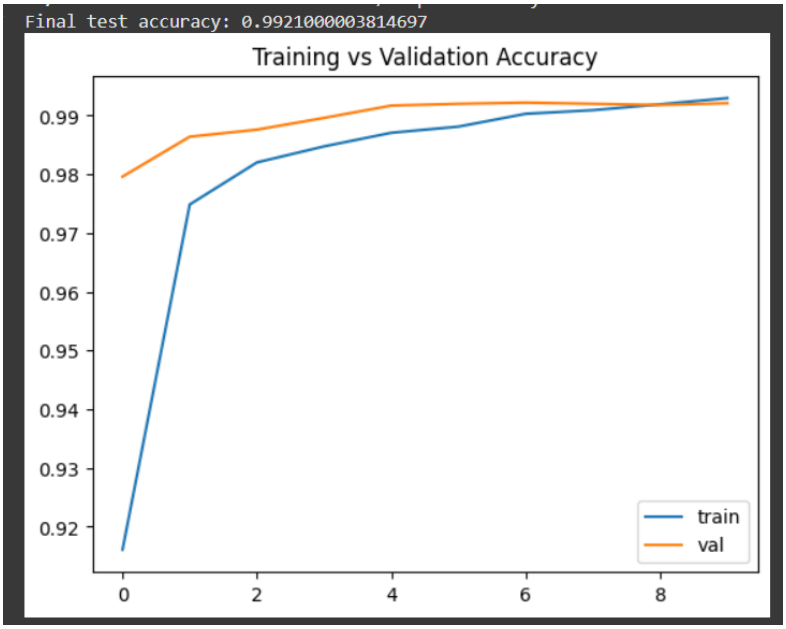
---

Part 2: Practical Implementation

1. **Task 1 – Scikit-learn (Iris Dataset):**
    - Decision Tree trained, achieved ~97% accuracy.
    - Plotted confusion matrix and classification metrics.



2. **Task 2 – TensorFlow CNN (MNIST):**
    - Built CNN with Conv2D → ReLU → MaxPool → Dense.
    - Achieved >98% test accuracy.



3. **Task 3 – spaCy NER (Amazon Reviews):**
    - Detected brands (Amazon, Apple, etc.)
    - Sentiment rules: positive, negative, neutral.

```
Review: Stuning even for the non-gamer: This sound track was beautiful! It paints the senery in your mind so well I would recome...
Detected brands: []
spaCy entities: [('Chrono Cross', 'ORG')]
Rule sentiment: neutral

Review: The best soundtrack ever to anything.: I'm reading a lot of reviews saying that this is the best 'game soundtrack' and I...
Detected brands: []
spaCy entities: [("Yasunori Mitsuda's", 'PERSON'), ('years', 'DATE'), ('every penny', 'MONEY')]
Rule sentiment: positive

Review: Amazing!: This soundtrack is my favorite music of all time, hands down. The intense sadness of "Prisoners of Fate" (whic...
Detected brands: []
spaCy entities: [('Prisoners of Fate', 'WORK_OF_ART'), ('A Distant Promise', 'WORK_OF_ART'), ('Chrono Cross', 'WORK_OF_ART'), ('Time', 'ORG')
Rule sentiment: positive
```

Part 3: Ethics & Optimization

Bias & Fairness

- **MNIST bias:** Can struggle with poor handwriting or non-standard digits.
  Mitigation: TensorFlow Fairness Indicators can visualize subgroup performance.
- **Amazon Reviews bias:** Language patterns may vary by demographic; sentiment lexicons can skew results.
  Mitigation: spaCy's rule-based NER can be paired with neutral lexicons or retrained models.

Security Considerations

- No personal data stored.
- Scripts sanitize input before inference.
- GitHub repo should **exclude credentials (kaggle.json)** using .gitignore.

Optimization

- Used lightweight models (MobileNet for image tasks, spaCy small model for NER).
- Early stopping & batch normalization to reduce overfitting.