

# AI Tools & Frameworks Assignment Report

---

## Part 1: Theoretical Understanding

### Q1: Differences Between TensorFlow and PyTorch

- **TensorFlow:** Static graph execution, better for production deployment (e.g., TensorFlow Lite, TensorFlow.js).
- **PyTorch:** Dynamic graph execution, easier debugging and more pythonic.
- **Use Case:** Choose TensorFlow for scalable deployment and PyTorch for rapid experimentation and research.

### Q2: Use Cases for Jupyter Notebooks

1. Interactive experimentation with data and models.
2. Step-by-step documentation for educational tutorials or demonstrations.

### Q3: spaCy vs. Python String Operations

- **spaCy** provides tokenization, named entity recognition (NER), and part-of-speech tagging.
- Unlike `str.split()` or regex, spaCy uses trained models, making it more accurate and context-aware.

### Comparative Analysis: Scikit-learn vs. TensorFlow

Feature	Scikit-learn	TensorFlow
Focus	Classical ML	Deep Learning (DL)
Beginner Friendly	Very easy to use	Steeper learning curve
Community Support	Mature, widely used	Huge, especially in DL field

---

## Part 2: Practical Implementation

### Task 1: Scikit-learn – Iris Dataset

- Preprocessed data by encoding labels.
- Trained a **Decision Tree Classifier**.
- Achieved high accuracy (~95%).
- Evaluated with accuracy, precision, and recall.

 Code: `ml/iris_classifier.ipynb`

## Task 2: TensorFlow – MNIST Dataset

- Built a **CNN** with 2 convolutional layers and dense layers.
- Achieved test accuracy >95%.
- Visualized predictions on 5 samples.

 Code: `dl/mnist_cnn_tensorflow.py`

## Task 3: spaCy – Amazon Reviews

- Used `en_core_web_sm` for NER.
- Rule-based sentiment analysis using keyword matching.
- Extracted product names and classified sentiment.

 Code: `nlp/spacy_ner_sentiment.py`

---



## Part 3: Ethics & Optimization

### Ethical Considerations

- **MNIST Bias:** Risk of biased handwriting recognition across cultures.
- **Amazon Reviews:** Incomplete entity detection and sentiment misunderstanding.
- **Mitigation:** Use Fairness Indicators and advanced NLP models like transformers.

### Debugging Challenge Example

**Issue:** Used `categorical_crossentropy` with integer labels.

**Fix:**

```
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy')
```

---



## Conclusion

This assignment helped reinforce the practical skills and ethical considerations involved in AI development. Each team member contributed to code, theory, and presentation, ensuring balanced participation and deeper understanding of real-world AI tooling.