

# AI Tools and Applications - Final Report

## 1. Theoretical Questions

### Q1: Difference Between TensorFlow and PyTorch

TensorFlow uses static computation graphs while PyTorch uses dynamic graphs. TensorFlow is better for production environments,

while PyTorch is preferred in research due to its ease of use and debugging.

### Q2: Two Use Cases for Jupyter Notebooks

1. Data exploration and visualization for AI experiments.
2. Collaborative sharing and execution of code for prototyping models.

### Q3: How spaCy Enhances NLP Compared to Python String Operations

spaCy provides pre-trained NLP pipelines that can extract entities, analyze grammar, and tokenize language effectively.

It outperforms basic string operations by understanding context and structure in text.

### Q4: Compare Scikit-learn vs TensorFlow

Scikit-learn is used for classical machine learning like decision trees and SVMs. TensorFlow is for deep learning and neural networks.

Scikit-learn is easier for beginners, while TensorFlow has more deployment features and scalability.

## 2. Practical Tasks

Task 1: Iris Classification (Scikit-learn)

- Preprocessed data from Iris dataset.
- Trained a Decision Tree Classifier.
- Evaluated using accuracy, precision, and recall.

# AI Tools and Applications - Final Report

## Task 2: MNIST Digit Recognition (TensorFlow)

- Built a CNN model.
- Achieved 99% test accuracy.
- Visualized predictions on 5 images.

## Task 3: NLP with spaCy (Amazon Reviews)

- Extracted named entities using spaCy.
- Performed rule-based sentiment analysis.

### 3. Ethical Reflection

While building AI models for digit recognition and text sentiment, we identified potential biases in dataset composition and processing techniques. MNIST lacks handwriting diversity, and our sentiment rules may overlook context in user language.

In future work, we'll explore fairness indicators and contextual NLP tools to improve ethical AI outcomes.

### 4. Debugging Challenge

A TensorFlow model was originally created with the wrong number of output classes and an incompatible loss function. We corrected the model by changing the Dense output layer to 10 neurons and using `sparse_categorical_crossentropy` loss.