Mastering the AI Toolkit

AI Tools Assignment Report

Team Members:

- Lamech Moses (+254111988590)
- Stephen Oloo (+254716631667)

Submission Format:

- GitHub repository (code + PDF report)
- Community article (this PDF uploaded)
- 3-minute project presentation (to be uploaded separately)

Part 1: Theoretical Understanding (30%)

1. Short Answer Questions

Q1: Explain the primary differences between TensorFlow and PyTorch. When would you choose one over the other?

*Ans: **TensorFlow** is developed by Google and is known for its production-readiness and scalability. It uses static computation graphs with eager execution support, making it ideal for deployment across platforms like mobile, embedded systems, and TensorFlow Serving while **PyTorch**, developed by Facebook, emphasizes dynamic computation graphs, offering a more Pythonic and intuitive approach. This makes it more flexible and popular in academic research and prototyping.*

*Choose **TensorFlow** when you need production-grade deployment, tools like TensorFlow Lite, or integration with TPUs.*
*Choose **PyTorch** for faster experimentation, better debugging, and a more natural coding style.*

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

Interactive Prototyping and Visualization:
Jupyter Notebooks allow data scientists to write, test, and debug code in cells. This makes it easier to iterate quickly on AI models while also visualizing performance using libraries like Matplotlib or Seaborn.

Reproducible Research and Reporting:
Jupyter notebooks can combine code, visuals, and markdown in a single document, making them perfect for sharing AI experiments, documenting findings, and enabling others to reproduce results.

Q3: How does spaCy enhance NLP tasks compared to basic Python string operations?
*Ans: While basic Python string methods can perform simple text manipulations like splitting or replacing words, they lack linguistic intelligence.**spaCy** is a powerful NLP library that enhances text analysis by providing Tokenization, Part-of-Speech tagging, Named Entity Recognition (NER), and Dependency Parsing out of the box and built-in language models that understand grammar, context, and relationships between words.*

This makes spaCy far more suitable for tasks like entity extraction, intent classification, and building chatbots — where understanding structure and semantics is essential.

2. Comparative Analysis

Criteria	Scikit-learn	TensorFlow
Target Applications	Classical ML models e.g SVM And Decision Trees.	Deep learning (ANNs, CNNs, and RNN)
Ease of Use	Very beginner-friendly	Steeper learning curve
Community Support	Strong, especially for academic tasks	Massive, especially in production

Part 2: Practical Implementation (40%)

Task 1: Classical Machine Learning with Scikit-learn

Description:

We implemented a Decision Tree Classifier on the Iris dataset using Scikit-learn. The workflow included loading the dataset, preprocessing (label encoding), splitting the data into training and test sets, training the model, and evaluating its performance.

Results:
- Accuracy: 1.00
- Precision: 1.00
- Recall: 1.00

Comment:
This perfect performance confirms the classifier is well-suited for distinguishing the three iris species in this dataset.

Task 2: Deep Learning with TensorFlow

Description:
We developed a Convolutional Neural Network (CNN) using TensorFlow and Keras to classify MNIST handwritten digits. The model architecture includes convolutional and pooling layers followed by dense layers. We trained the model for 5 epochs, achieving high accuracy on the test set.

Key Metrics:
Test Accuracy Achieved: ~99%

Output Screenshots:
`mnist_predicted_7_actual_7.png`
`mnist_predicted_1_actual_1.png`

Comment:
The CNN model successfully learned digit patterns, exceeding the 95% accuracy target and demonstrating strong generalization to unseen data.

Task 3: NLP with spaCy

Description:
We used spaCy to perform Named Entity Recognition (NER) on sample Amazon product reviews. A simple rule-based approach was implemented for sentiment analysis (detecting positive or negative tone based on keywords).

Example Outputs:

> Review: *I love my new Samsung Galaxy phone. Battery life is amazing!*
> Entities: Samsung Galaxy (ORG)
> Sentiment: Positive

> Review: *This Apple Watch is terrible. It stopped working after a week.*
> Entities: Apple Watch (ORG)
> Sentiment: Negative

> Review: *The Sony headphones are decent for the price.*
> Entities: Sony (ORG)
> Sentiment: Positive

> Review: *I hate this cheap charger, it's useless.*
> Entities: None
> Sentiment: Negative

Comment:
This approach extracts product and brand names while classifying sentiment to support product analysis use cases.

Part 3: Ethics & Optimization (10%)

1. Ethical Considerations

Potential Biases:
MNIST classifier: Possible overfitting or class imbalance may cause misclassification, particularly for under-represented digits.
NER for Amazon Reviews: Limited training data may exclude non-English product names, slang, or local brands, resulting in biased or incomplete extraction.

Mitigation Strategies:
Use **TensorFlow Fairness Indicators** to evaluate model fairness across digit classes.
Expand spaCy's rule-based patterns and training data to better handle diverse, multilingual inputs and improve inclusivity.

2. Troubleshooting Challenge

Example TensorFlow bug fix:
Incorrect loss function can cause shape errors:

Authors

Prepared and submitted by:

- **Lamech Moses**
- **Stephen Oloo**