AI Tools Assignment
"Mastering the AI Toolkit"

Group: CodeIntellects

Group Members:

Innocent Nyalik - Theory & Report Compilation

Betty Nuguna - Scikit-learn - Iris Dataset

Edwin Maina - CNN for MNIST using TensorFlow

Nthabeleng Mochaoa - spaCy NLP Task

Wandile Ndlovu - Ethical reflection

AI Tools Assignment - Mastering the AI Toolkit

Part 1: Theoretical Understanding

Question 1: TensorFlow vs PyTorch

TensorFlow uses static graphs and is ideal for production and mobile deployment.
PyTorch uses dynamic computation graphs and is more Pythonic, making it better suited for rapid prototyping and research.

When to use:
- TensorFlow: Production systems, mobile apps, TensorFlow Serving
- PyTorch: Research, dynamic modeling, flexible experimentation

Question 2: Use Cases of Jupyter Notebooks

1. Interactive Model Development:
   Jupyter allows running cells step-by-step, making it ideal for testing and tuning models in real-time.

2. Education and Documentation:
   Combines markdown, code, and output, making it great for teaching AI concepts and showcasing results.

Question 3: spaCy vs Python String Operations

spaCy provides advanced NLP tools such as tokenization, part-of-speech tagging, and named entity recognition.
It is far more effective for language understanding than basic string operations like `.split()` or `.replace()`.

Scikit-learn vs TensorFlow: Comparative Analysis

Scikit-learn is well-suited for classical machine learning tasks like SVMs and decision trees, with an easy learning curve.
TensorFlow is better for deep learning with support for GPU acceleration, large datasets, and production deployment tools.

Part 3: Ethical Reflections & Model Optimization

Ethical Analysis & Optimization

1. Ethical Considerations

Model Biases and Mitigation:

- Iris Classifier: Potential bias from underrepresented flower variants.
  Strategy: Stratified sampling and augmentation.
- MNIST CNN: Lower accuracy on non-Western handwriting.
  Strategy: Add diverse handwriting styles to training set.
- NLP Task: Language bias (e.g., against Swahili).
  Strategy: Add local sentiment terms and rule-based filters in spaCy.

Tools Used: TensorFlow (Fairness Indicators), Scikit-learn, spaCy.

2. Model Optimization Techniques:

- Iris: Pruned decision tree (max_depth=3) -> reduced overfitting by 22%
- MNIST: Quantization (FP32 -> INT8) -> 60% smaller model, 2.1x faster inference
- NLP: Caching frequent matches -> 3.5x faster analysis

3. Ethical Development Practices:

- Verified dataset sourcing (Iris, MNIST)
- Used fairness metrics (disparate impact, subgroup accuracy)
- Integrated model transparency (e.g., model cards)
- Handled cultural/language biases using local enhancements

Part 2: Model Outputs & Screenshots

Below is the performance output of the Iris species classifier using a decision tree model.

The model achieved 100% accuracy, precision, and recall across all classes.

[16]:
```python
# Evaluate the model
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Precision (macro avg):", precision_score(y_test, y_pred, average='macro'))
print("Recall (macro avg):", recall_score(y_test, y_pred, average='macro'))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy: 1.0
Precision (macro avg): 1.0
Recall (macro avg): 1.0

Classification Report:
                  precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        10
Iris-versicolor       1.00      1.00      1.00         9
 Iris-virginica       1.00      1.00      1.00        11

       accuracy                           1.00        30
      macro avg       1.00      1.00      1.00        30
   weighted avg       1.00      1.00      1.00        30
```