**AI Tools Assignment: AI Tools and Applications**  
**Theme**: *"Mastering the AI Toolkit"* 🛠️🧠

**Objective & Guidelines**

This assignment evaluates your understanding of **AI tools/frameworks** and their real-world applications through a mix of theoretical and practical tasks. You’ll demonstrate proficiency in selecting, implementing, and critically analyzing AI tools to solve problems.

The assignment has three parts: theory, which tests the theoretical understanding of AI Tools; Practical, which tests the implementation skills of AI tools; and Ethics and optimization, which tests ethical AI development and AI software optimization skills.

The assignment should be handled as group work formed by 3-5 people. This is to enhance teamwork and AI engineering collaboration.

**Submission Guidelines**

1. **Code**: Well-commented scripts/Jupyter notebooks. (To be submitted on GitHub)
2. **Report**: A PDF with:
   * Answers to theoretical questions.
   * Screenshots of model outputs (e.g., accuracy graphs, NER results).
   * Ethical reflection.
   * *(Share this PDF as an article in the Community, Peer Group Review and include it in the GitHub Repo)*
3. **Presentation**: Create 3-minute video explaining your approach. (All members in the group should participate in the video. Share the video on the Community platform)

**Grading Rubric**

| **Criteria** | **Weight** |
| --- | --- |
| Theoretical Accuracy | 30% |
| Code Functionality & Quality | 40% |
| Ethical Analysis | 15% |
| Creativity & Presentation | 15% |

**Tools & Resources**

* **Frameworks**: TensorFlow, PyTorch, Scikit-learn, spaCy.
* **Platforms**: Google Colab (free GPU), Jupyter Notebook.
* **Datasets**: Kaggle, TensorFlow Datasets.

**Why This Matters**

* **Real-World Impact**: These tools power industries from healthcare to finance.
* **Skill Validation**: Employers seek proficiency in TensorFlow, PyTorch, and Scikit-learn.

**Deadline**: 7 days. Showcase your AI toolkit mastery! 🚀

**Need Help?**

* Use official documentation: [TensorFlow](https://www.tensorflow.org/), [PyTorch](https://pytorch.org/), [spaCy](https://spacy.io/).
* Post questions on the LMS Community with #AIToolsAssignment.

**Pro Tip**: *Test code incrementally—small wins lead to big successes! 💡*

**Part 1: Theoretical Understanding (40%)**

**1. Short Answer Questions**

* **Q1**: Explain the primary differences between TensorFlow and PyTorch. When would you choose one over the other?
* **Q2**: Describe two use cases for Jupyter Notebooks in AI development.
* **Q3**: How does spaCy enhance NLP tasks compared to basic Python string operations?

**2. Comparative Analysis**

* Compare Scikit-learn and TensorFlow in terms of:
  + Target applications (e.g., classical ML vs. deep learning).
  + Ease of use for beginners.
  + Community support.

**Part 2: Practical Implementation (50%)**

**Task 1: Classical ML with Scikit-learn**

* **Dataset**: [Iris Species Dataset](https://www.kaggle.com/uciml/iris)
* **Goal**:
  1. Preprocess the data (handle missing values, encode labels).
  2. Train a **decision tree classifier** to predict iris species.
  3. Evaluate using accuracy, precision, and recall.
* **Deliverable**: Python script/Jupyter notebook with comments explaining each step.

**Task 2:** **Deep Learning with TensorFlow/PyTorch**

* **Dataset**: [MNIST Handwritten Digits](https://www.tensorflow.org/datasets/catalog/mnist)
* **Goal**:
  1. Build a **CNN model** to classify handwritten digits.
  2. Achieve >95% test accuracy.
  3. Visualize the model’s predictions on 5 sample images.
* **Deliverable**: Code with model architecture, training loop, and evaluation.

**Task 3: NLP with spaCy**

* **Text Data**: User reviews from [Amazon Product Reviews](https://www.kaggle.com/bittlingmayer/amazonreviews).
* **Goal**:
  1. Perform **named entity recognition (NER)** to extract product names and brands.
  2. Analyze sentiment (positive/negative) using a rule-based approach.
* **Deliverable**: Code snippet and output showing extracted entities and sentiment.

**Part 3: Ethics & Optimization (10%)**

**1. Ethical Considerations**

* Identify potential biases in your MNIST or Amazon Reviews model. How could tools like **TensorFlow Fairness Indicators** or **spaCy’s rule-based systems** mitigate these biases?

**2. Troubleshooting Challenge**

* **Buggy Code**: A provided TensorFlow script has errors (e.g., dimension mismatches, incorrect loss functions). Debug and fix the code.

**Bonus Task (Extra 10%)**

* **Deploy Your Model**: Use **Streamlit** or **Flask** to create a web interface for your MNIST classifier. Submit a screenshot and a live demo link.

Connecting to kernel: Python 3.13.0: Activating Python Environment 'Python 3.13.0'