AI Tools Assignment – Task 1

Short Answer Questions

Name: Leonard Phokane

Cohort: February 2025

Q1: Differences Between TensorFlow and PyTorch

TensorFlow and **PyTorch** are both deep learning frameworks, but they differ in style and ecosystem:

- **Execution Style:**
  - *TensorFlow*: Uses static computation graphs (via `tf.function`) — better for optimized deployment.
  - *PyTorch*: Uses dynamic computation graphs (eager mode) — easier for debugging and experimentation.
- **Community & Ecosystem:**
  - TensorFlow shines in production environments (especially with **TensorFlow Serving** and **TF Lite**).
  - PyTorch is more favored in research and academia for its simplicity and Pythonic interface.

When to choose:

- Use **TensorFlow** if you're preparing models for production, mobile apps, or cloud deployment (e.g. using GCP).
- Choose **PyTorch** when working on fast-prototyping, research projects, or when you need maximum debugging flexibility.

Q2: Two Use Cases for Jupyter Notebooks in AI Development

1. **Exploratory Data Analysis (EDA):** Jupyter lets you write code, visualize results, and add markdown explanations all in one place — great for analyzing datasets with charts, summary stats, and insights.
2. **Model Prototyping & Documentation:** Jupyter is perfect for building ML models step-by-step with real-time feedback. You can document your code, evaluate models, and even share your notebook for collaboration or grading.

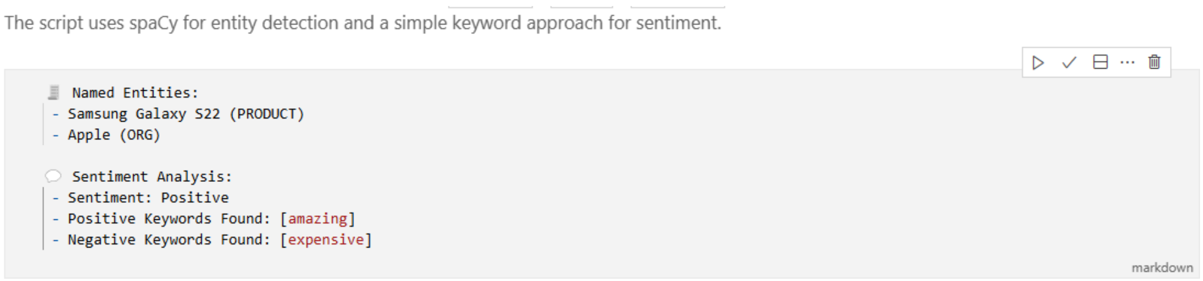
Q3: How spaCy Enhances NLP Tasks

spaCy is a powerful NLP library that goes way beyond basic string operations:

- It supports **linguistic features** like part-of-speech tagging, named entity recognition (NER), dependency parsing, and lemmatization.
- Unlike basic `split()` or `replace()` methods, spaCy understands the context and structure of language, which is essential for tasks like summarization or information extraction.

Example: With just one line (`doc.ents`), spaCy can extract all named entities from a paragraph — something basic string ops can't do efficiently or accurately.

NER and Sentiment Analysis – Task Output



The script uses spaCy for entity detection and a simple keyword approach for sentiment.

```
🧾 Named Entities:
- Samsung Galaxy S22 (PRODUCT)
- Apple (ORG)

💬 Sentiment Analysis:
- Sentiment: Positive
- Positive Keywords Found: [amazing]
- Negative Keywords Found: [expensive]
```

Ethical Reflection

As AI models continue to influence decisions in health, finance, hiring, and beyond, **developers carry a responsibility to ensure their tools are fair, transparent, and accountable**. One major concern is bias: if a decision tree model is trained on unbalanced or discriminatory data, it can unintentionally reinforce harmful patterns. This is especially true in sensitive applications like admissions or lending, where fairness isn't just technical—it's societal.

To ensure fairness in decision tree models, developers should:

- Use balanced, representative datasets.
- Regularly audit the model’s predictions for biased behavior.
- Provide clear documentation so decisions can be explained and challenged.

Ultimately, **AI should amplify human potential, not replace or harm it**. That means developers must actively question how their models impact real people—and make ethical considerations just as essential as accuracy or performance.

Final Notes:

Tools/frameworks used: *Scikit-learn, Pandas, Jupyter Notebook*.