Theoretical Understanding

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

TensorFlow and PyTorch are two of the most widely used open-source deep learning frameworks in the machine learning community. While they serve similar purposes, each offers distinct features, workflows, and ecosystems that make them suitable for different use cases. Understanding their primary differences is crucial when deciding which to use for a particular project—whether it's cutting-edge research, rapid prototyping, or deploying models at scale.

Feature	TensorFlow	PyTorch
Developer	Google	Meta
Computation type	Static graphs	Dynamic graphs
Ease of use	Steeper learning curve	pythonic
Community support	Large and mature	Rapidly growing
Visualization	Built-in Tensor board	Tensor board among other tools
Deployment	TensorFlow Serving, TFLite, TensorFlow.js	Torch Script, ONNX
Preferred use	Production, mobile, and web deployment	Research, experimentation, fast prototyping

Choose TensorFlow when:

- You need robust tools for production, deployment, and cross-platform support.
- You're working on mobile or web apps (e.g., TFLite, TensorFlow.js).
- Visualization with TensorBoard is important.

Choose PyTorch when:

- You're doing academic research, NLP, or computer vision.
- You prefer quick debugging and a more Pythonic coding style.
- You need dynamic computation graphs for flexibility.

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

Exploratory Data Analysis (EDA).

Jupyter Notebooks are ideal for exploring datasets, visualizing data distributions, and identifying patterns or anomalies. Data scientists use them to write code, visualize outputs (like charts and graphs), and document findings—all in one interactive environment.

Model Prototyping and Experimentation.

Jupyter allows for rapid prototyping of machine learning models. Developers can test different algorithms, adjust hyperparameters, and view real-time results, making it easier to iterate and refine AI models before moving to production.

Q3: How does spaCy enhance NLP tasks compared to basic Python string operations?

spaCy significantly enhances NLP tasks by providing **advanced, efficient, and linguistically aware tools**, which go far beyond basic Python string methods like `.split()`, `.replace()`, or `.lower()`.

The following are key enhancements:

Linguistic Awareness.
spaCy understands the structure of language. It offers features like **tokenization**, **part-of-speech tagging**, **named entity recognition (NER)**, and **dependency parsing**, which are essential for truly understanding text context—something basic string functions cannot do.

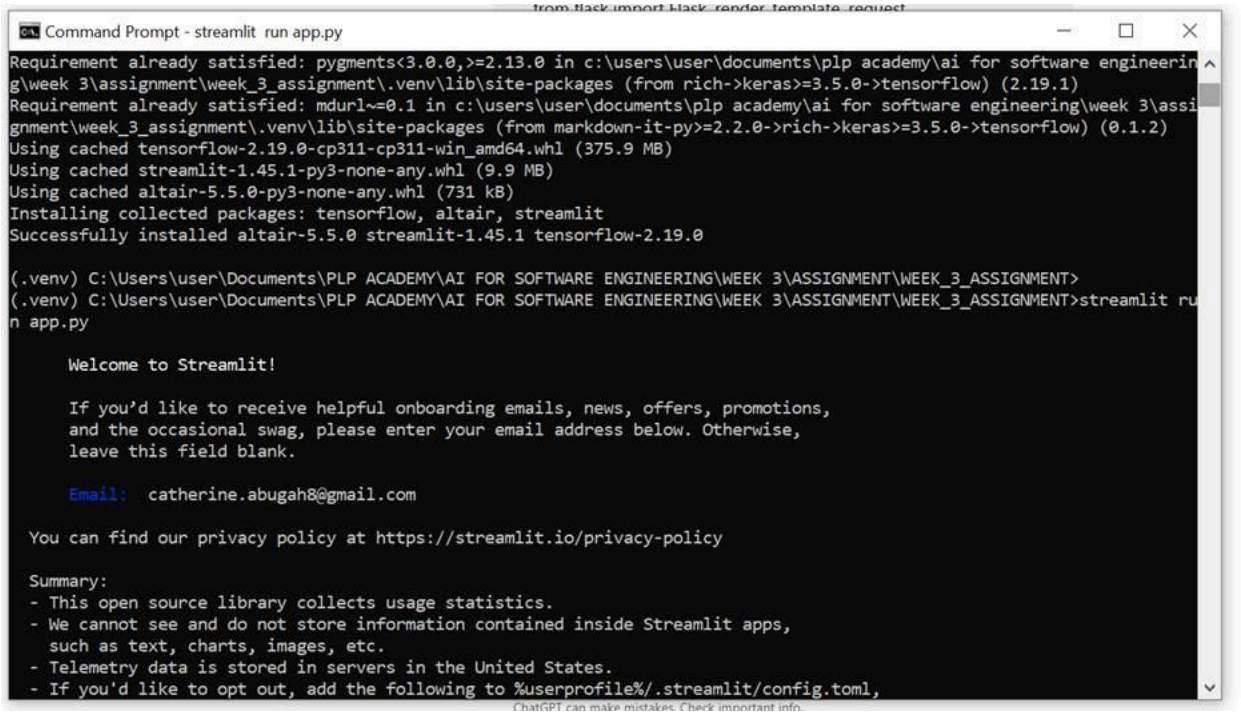
Speed and Efficiency.
spaCy is designed for industrial-strength NLP. It processes large volumes of text **faster and more accurately** than custom code written with standard Python operations.

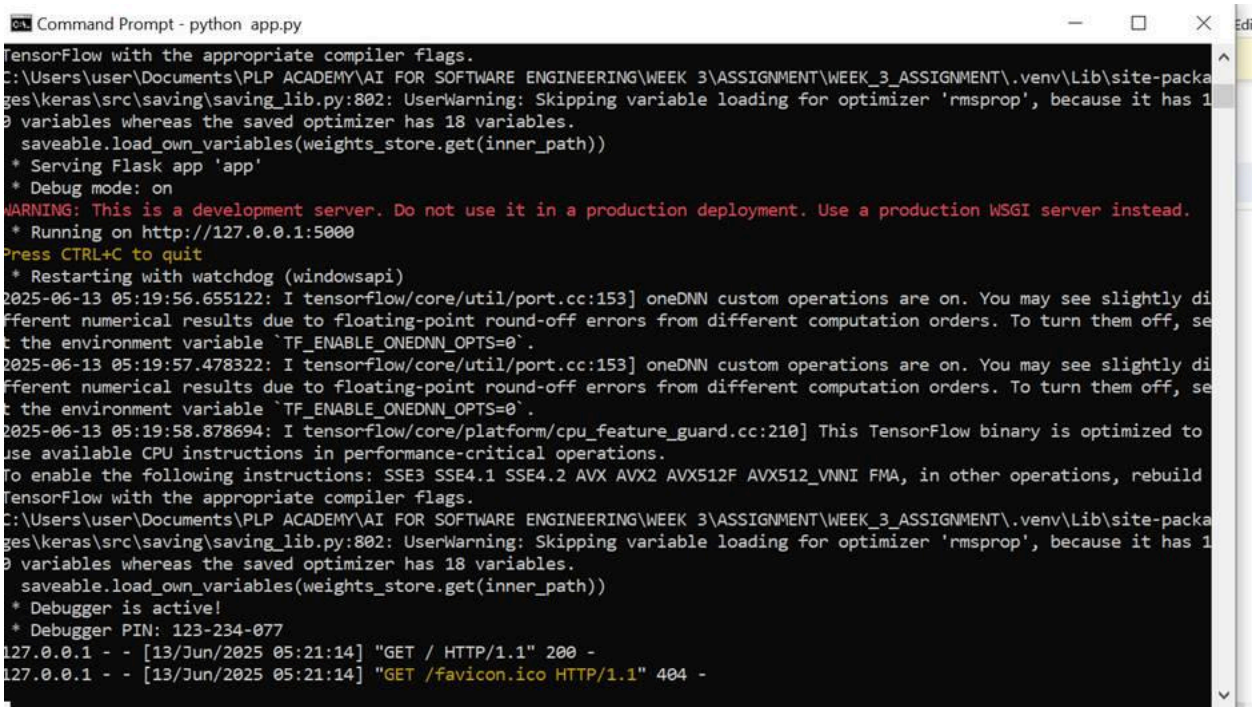
Built-in Models and Pipelines.
spaCy comes with **pretrained models** that can recognize common entities (like names, dates, organizations), which would require substantial manual effort to handle with basic string operations.

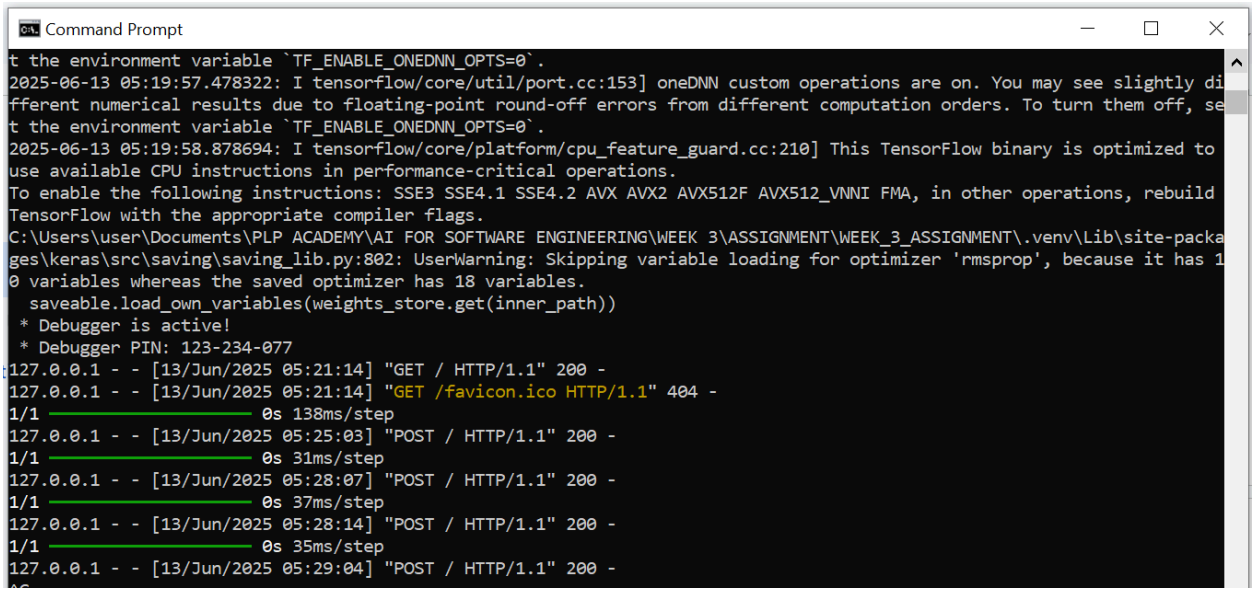
Comparative Analysis between Scikit-learn and TensorFlow

1. **Scikit-learn** is primarily used for classical machine learning tasks such as classification, regression, clustering, and dimensionality reduction. It is highly suitable for structured (tabular) data and includes algorithms like decision trees, random forests, linear models, and support vector machines. In contrast, **TensorFlow** is designed for building and training deep learning models and is widely used for complex tasks like image recognition, natural language processing, and sequence modeling.
2. For beginners, Scikit-learn is much easier to get started with due to its simple and consistent API, making it ideal for those new to machine learning. TensorFlow, while more powerful and flexible, has a steeper learning curve—though the integration of Keras in TensorFlow 2.x has made it more accessible by simplifying model building and training.
3. In terms of community support, both libraries have strong followings. Scikit-learn has robust support within academic and traditional data science communities. TensorFlow, backed by Google, has a vast global community, frequent updates, and rich learning resources, especially for deep learning and AI research.

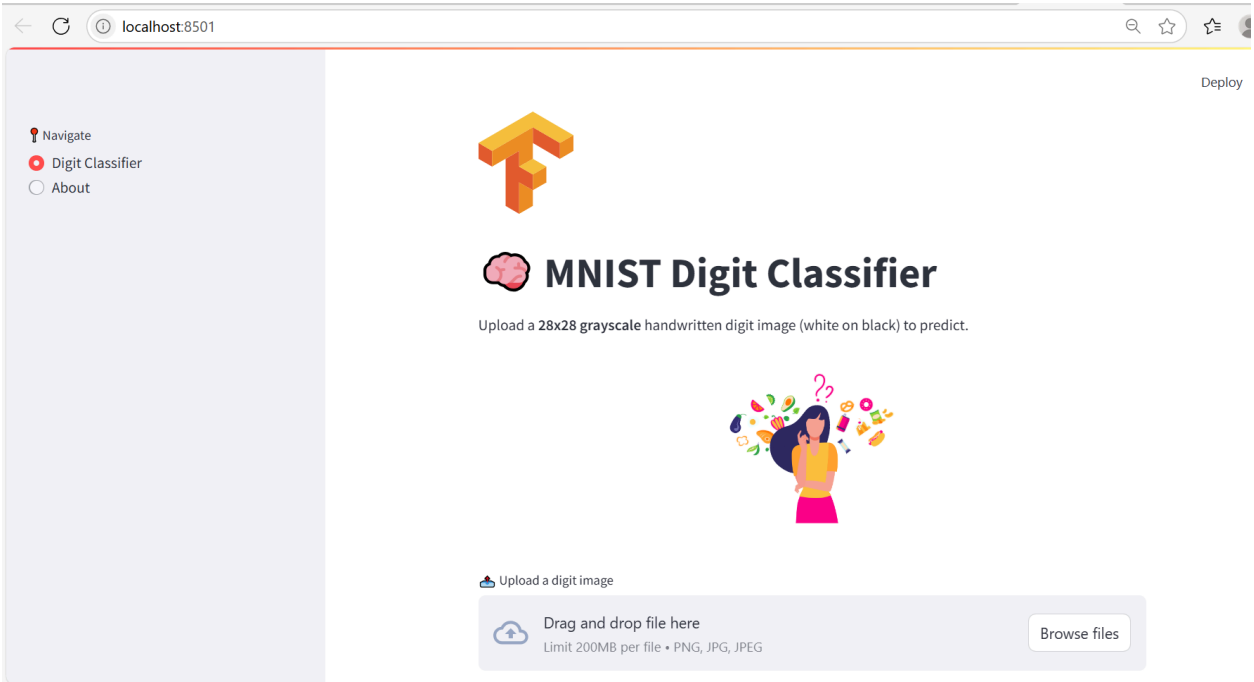
SCREENSHOTS

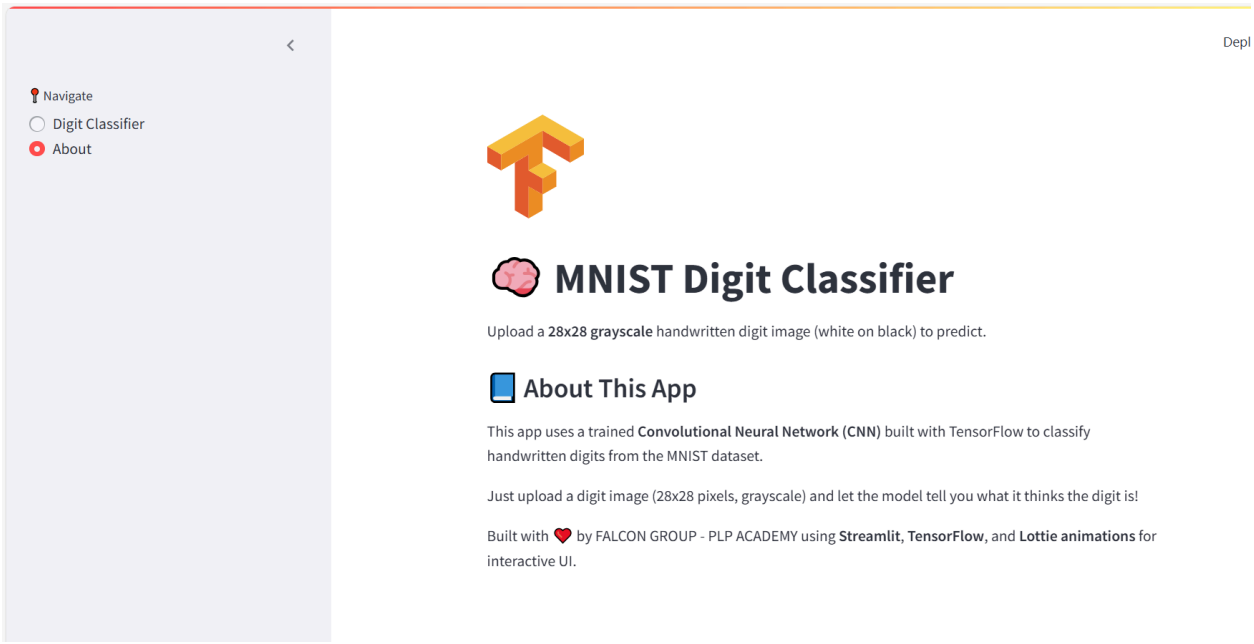





Command Prompt
t the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-06-13 05:19:57.478322: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly di
fferent numerical results due to floating-point round-off errors from different computation orders. To turn them off, se
t the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-06-13 05:19:58.878694: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to
use available CPU instructions in performance-critical operations.
To enable the following instructions: SSE3 SSE4.1 SSE4.2 AVX AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebuild
TensorFlow with the appropriate compiler flags.
C:\Users\user\Documents\PLP ACADEMY\AI FOR SOFTWARE ENGINEERING\WEEK 3\ASSIGNMENT\WEEK_3_ASSIGNMENT\.venv\Lib\site-packa
ges\keras\src\saving\saving_lib.py:802: UserWarning: Skipping variable loading for optimizer 'rmsprop', because it has 1
0 variables whereas the saved optimizer has 18 variables.
saveable.load_own_variables(weights_store.get(inner_path))
* Debugger is active!
* Debugger PIN: 123-234-077
127.0.0.1 - - [13/Jun/2025 05:21:14] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [13/Jun/2025 05:21:14] "GET /favicon.ico HTTP/1.1" 404 -
1/1 0s 138ms/step
127.0.0.1 - - [13/Jun/2025 05:25:03] "POST / HTTP/1.1" 200 -
1/1 0s 31ms/step
127.0.0.1 - - [13/Jun/2025 05:28:07] "POST / HTTP/1.1" 200 -
1/1 0s 37ms/step
127.0.0.1 - - [13/Jun/2025 05:28:14] "POST / HTTP/1.1" 200 -
1/1 0s 35ms/step
127.0.0.1 - - [13/Jun/2025 05:29:04] "POST / HTTP/1.1" 200 -

Page


localhost:8501
Deploy
Navigate
Digit Classifier
About
MNIST Digit Classifier
Upload a 28x28 grayscale handwritten digit image (white on black) to predict.
Upload a digit image
Drag and drop file here
Limit 200MB per file • PNG, JPG, JPEG
Browse files



Uploaded image then here is the Prediction