WEEK 3 AI ASSIGNMENT

PART 1: THEORETICAL UNDERSTANDING

Q1. PRIMARY DIFFERENCE BETWEEN TENSORFLOW AND PYTORCH

Tensorflow is a powerful model for building and training deep learning models like image recognition, speech processing while pytorch is similar to tensorflow in that it is a deep learning library but it is simpler and more flexible for experimentation and research.

WHEN WOULD YOU CHOOSE ONE OVER THE OTHER?

Pytorch is best for research and experimentation so I would choose it where flexibility and ease of debugging matter.

Tensorflow is best for production and deployment so I would choose it where scalability and integration with serving or mobile platforms are needed.

Q2. DESCRIBE 2 USE CASES FOR JUPYTER NOTEBOOK IN AI DEVELOPMENT

Exploratory data analysis. It allows developers to load datasets, visualize distributions, correlations and summary statistics using tools like pandas and matplotlib interactively.

Machine learning experimentation and reporting. Developers can train models, visualize training results and document their process in the same environment.

Q3. HOW DOES SPACY ENHANCE NLP TASKS COMPARED TO BASIC PYTHON STRING OPERATIONS?

spaCy enhances NLP tasks by providing advanced language understanding .It can recognize parts of speech, named entities like brands or people and sentence structure whereas basic Python string operations can only handle simple text tasks like splitting or searching for words.

COMPARATIVE ANALYSIS

COMPARING SCIKIT LEARN AND TENSORFLOW

Feature	Scikit-learn	TensorFlow
Target Applications	Designed for classical machine learning (e.g., regression, decision trees, SVMs, clustering).	Built for deep learning and neural networks, including large-scale and complex models.
Ease of Use for Beginners	Easier to learn. It is simple, consistent API and quick to implement models.	Steeper learning curve. It requires understanding of neural network concepts and tensor operations.
Community Support	Large and mature community focused on traditional ML; extensive documentation and examples.	Massive global community with strong support from Google and deep learning researchers; many tutorials and tools available.

PART 2: PRACTICAL IMPLEMENTATION

TASK 1

Model Evaluation Results

1. Accuracy: 0.9333 (≈ 93.33%)

The model correctly predicted **93.33%** of all test samples.

Indicates strong overall performance.

2. Precision: 0.9333 (≈ 93.33%)

Precision measures how many samples predicted as a certain class were actually correct.

High precision means few false positives .The model rarely labels the wrong species.

3. Recall: 0.9333 (≈ 93.33%)

Recall measures how many of the actual samples of a class were correctly identified.

High recall means few false negatives .The model successfully captured most of the correct samples.

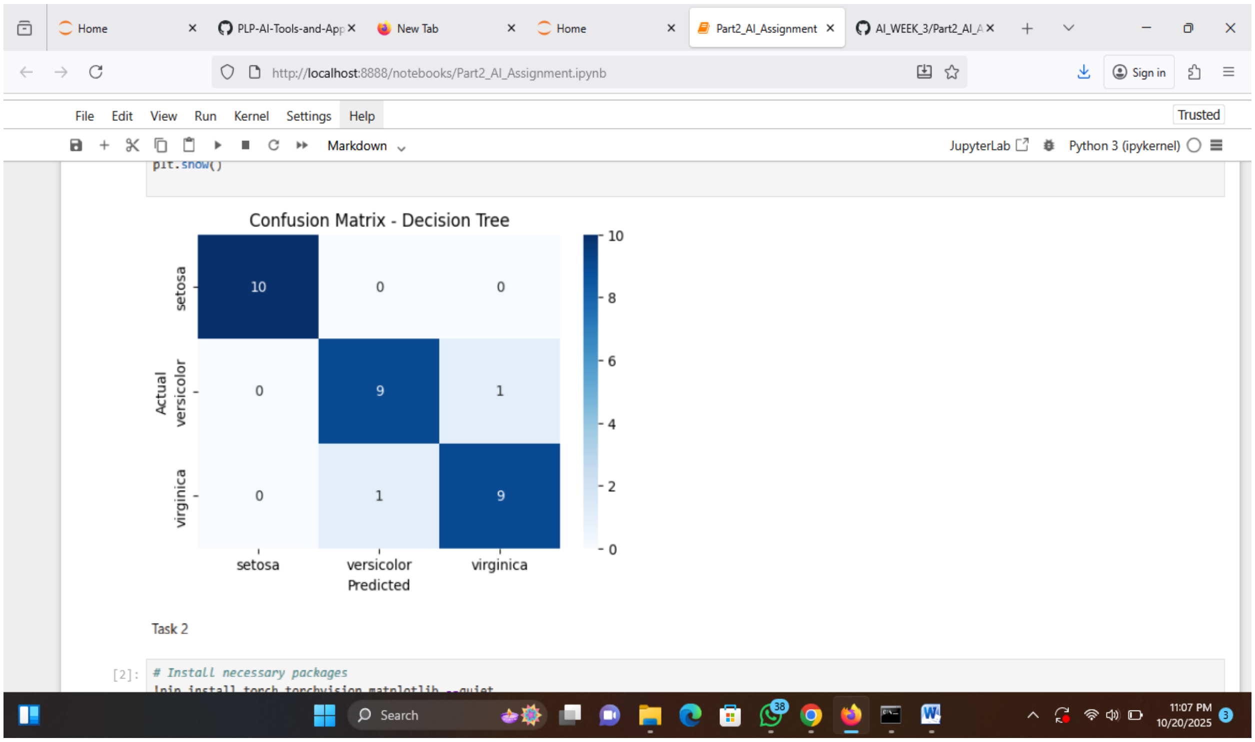
Classification Report Breakdown

Class	Precision	Recall	F1-score	Support
Setosa	1.00	1.00	1.00	10
Versicolor	0.90	0.90	0.90	10
Virginica	0.90	0.90	0.90	10

- **Setosa** was perfectly classified — 100% accuracy in predictions.
- **Versicolor** and **Virginica** show slight misclassifications, which is expected since their features overlap (especially petal length and width).
- Support indicates the number of test samples per class (10 each in this case).

Interpretation Summary

The model demonstrates strong and balanced performance across all metrics.
Minor misclassifications occur mainly between Versicolor and Virginica, which are closely related species in feature space.
The results suggest the model generalizes well and can accurately classify most flower samples in the dataset.



TASK 2

Interpretation of Training Loss

The CNN model was trained for 3 epochs on the MNIST dataset, and the training loss per epoch was as follows:

- **Epoch 1:** Loss = 0.1321
- **Epoch 2:** Loss = 0.0433
- **Epoch 3:** Loss = 0.0282

Analysis

- **Epoch 1 (0.1321):** At the start of training, the model began learning the basic patterns of handwritten digits, such as simple shapes and edges. The relatively low initial loss indicates that the CNN is already making mostly correct predictions.
- **Epoch 2 (0.0433):** The substantial decrease in loss shows that the model is learning more refined features, including loops, curves, and line orientations. This indicates rapid improvement in its classification ability.
- **Epoch 3 (0.0282):** The loss is very low, suggesting that the model is highly confident in its predictions and is classifying the training images correctly most of the time.

General Observations

The rapid decline in training loss demonstrates **fast convergence**, which is typical for MNIST because it is a relatively simple dataset.

The low final loss implies that the model has effectively learned the key features needed for accurate digit classification.

To confirm that the model generalizes well to new data, it is important to check the **test accuracy**, ensuring that the model is not overfitting the training data.

Interpretation of Test Accuracy

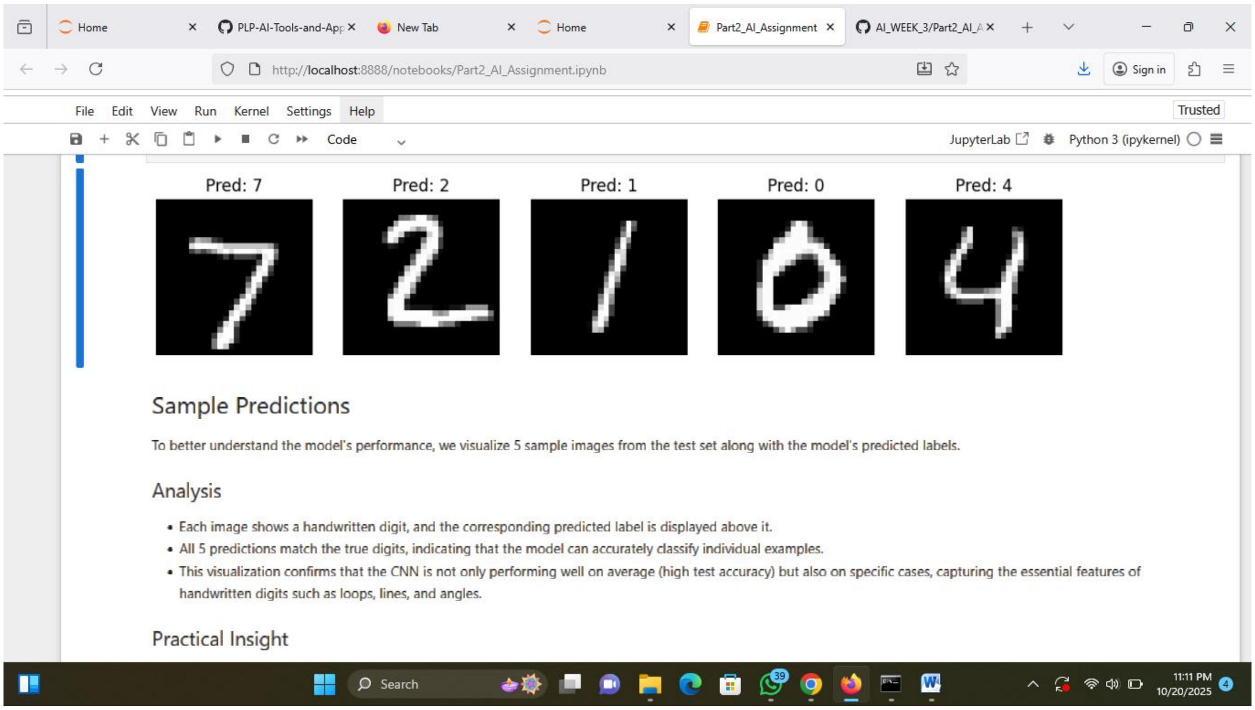
After training the CNN for 3 epochs, the model achieved a **test accuracy of 98.78%** on the MNIST dataset.

Analysis

- **High accuracy:** A test accuracy of 98.78% indicates that the model generalizes very well to unseen data and can correctly classify almost all handwritten digits in the test set.
- **Comparison with training loss:** The low training loss combined with high test accuracy shows that the model has learned meaningful patterns without overfitting.
- **Model performance:** Achieving accuracy above 95% is considered excellent for MNIST classification. This confirms that the CNN architecture and training procedure are effective.
- **Practical implication:** The model can reliably be used to recognize handwritten digits in real-world applications, demonstrating strong predictive capability.

Summary

Overall, the CNN is highly accurate and robust, achieving both low training loss and high test accuracy, which indicates successful learning of the features necessary for digit classification.



Sample Predictions

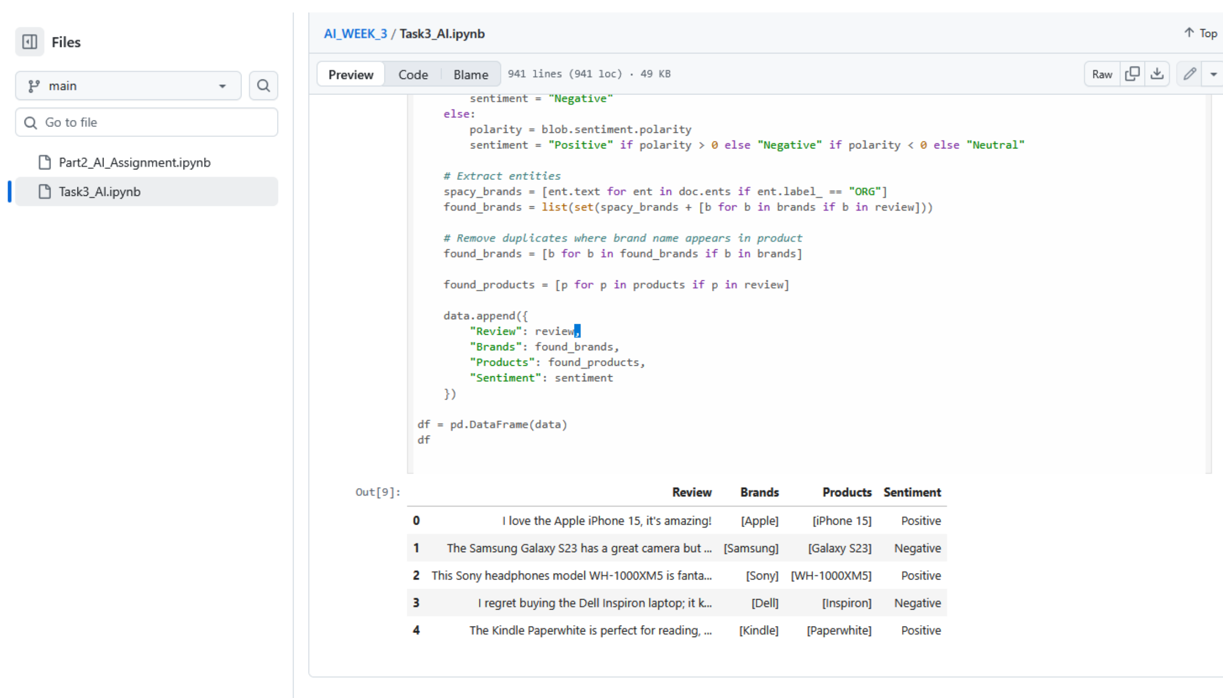
To better understand the model's performance, we visualize 5 sample images from the test set along with the model's predicted labels.

Analysis

- Each image shows a handwritten digit, and the corresponding predicted label is displayed above it.
- All 5 predictions match the true digits, indicating that the model can accurately classify individual examples.
- This visualization confirms that the CNN is not only performing well on average (high test accuracy) but also on specific cases, capturing the essential features of handwritten digits such as loops, lines, and angles.

Practical Insight

TASK 3




```
        sentiment = "Negative"
    else:
        polarity = blob.sentiment.polarity
        sentiment = "Positive" if polarity > 0 else "Negative" if polarity < 0 else "Neutral"

    # Extract entities
    spacy_brands = [ent.text for ent in doc.ents if ent.label_ == "ORG"]
    found_brands = list(set(spacy_brands + [b for b in brands if b in review]))

    # Remove duplicates where brand name appears in product
    found_brands = [b for b in found_brands if b in brands]

    found_products = [p for p in products if p in review]

    data.append({
        "Review": review,
        "Brands": found_brands,
        "Products": found_products,
        "Sentiment": sentiment
    })

df = pd.DataFrame(data)
df
```

Out[9]:

	Review	Brands	Products	Sentiment
0	I love the Apple iPhone 15, it's amazing!	[Apple]	[iPhone 15]	Positive
1	The Samsung Galaxy S23 has a great camera but ...	[Samsung]	[Galaxy S23]	Negative
2	This Sony headphones model WH-1000XM5 is fanta...	[Sony]	[WH-1000XM5]	Positive
3	I regret buying the Dell Inspiron laptop; it k...	[Dell]	[Inspiron]	Negative
4	The Kindle Paperwhite is perfect for reading, ...	[Kindle]	[Paperwhite]	Positive

PART 3: ETHICAL CONSIDERATION AND OPTIMIZATION

ETHICAL CONSIDERATIONS

IDENTIFY POTENTIAL BIASES IN YOURMNIST OR AMAZON REVIEWS MODEL. HOW COULD TOOLS LIKE TENSORFLOW FAIRNESS INDICATORS OR SPACY'S RULE-BASED SYSTEMS MITIGATE THESE BIASES?

Potential Biases:

MNIST model**:** Bias could occur if the model performs better on certain digits due to uneven training data distribution (e.g., more samples of "1" than "8").

Amazon Reviews model**:** Bias may appear if the dataset contains more positive reviews for certain brands (like Apple) and more negative ones for others (like Dell), leading to unfair sentiment predictions.

Mitigation Tools:

TensorFlow Fairness Indicators**:** This tool helps analyze model performance across subgroups (e.g., brand names). It can detect if the model treats one brand more favorably than another.

spaCy's rule-based systems**:** Custom rules can be created to handle biased language — for example, detecting overused sentiment words ("amazing," "terrible") that might unfairly influence sentiment classification.