MORAA ROBERT

MACHINE LEARNING WEEK 3 ASSIGNMENT

POWER LEARN PROJECT ARICA

1. Introduction

This project explores the practical use of modern **Artificial Intelligence (AI) tools and frameworks** to solve real-world machine learning and natural language processing problems.
The main goal was to understand how AI frameworks such as **TensorFlow**, **Scikit-learn**, and **spaCy** can be applied for different types of tasks from image classification and text analysis to evaluating model ethics and performance.

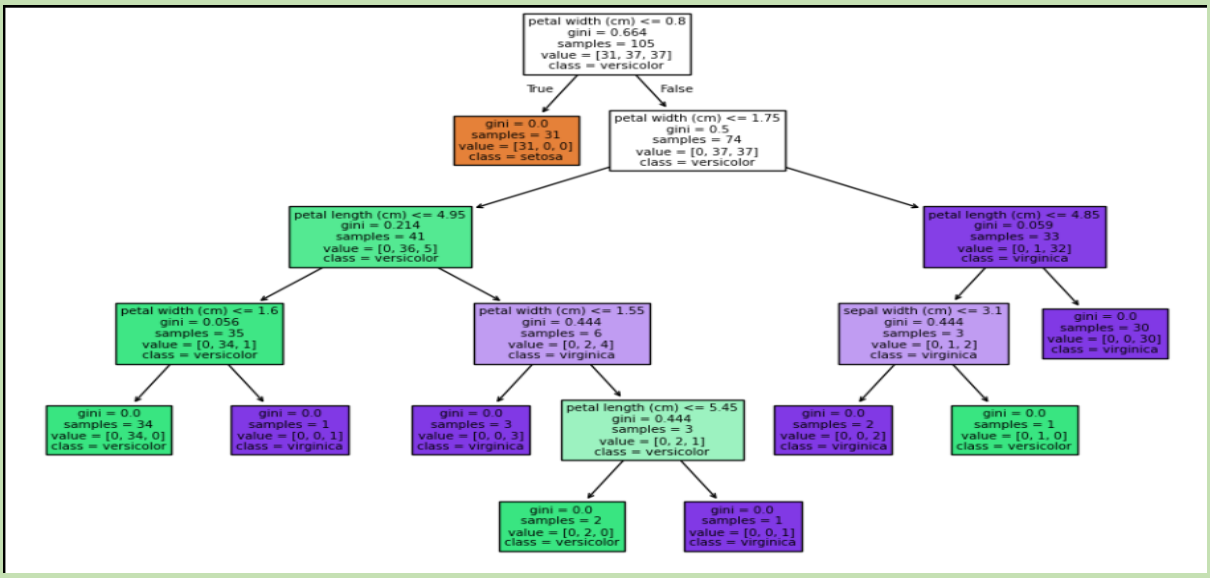
2. Tools and Frameworks Used

- **TensorFlow:** Used for building and training a Convolutional Neural Network (CNN) for image classification on the MNIST dataset.
- **Scikit-learn:** Used for implementing a classical machine learning model (Decision Tree Classifier) on the Iris dataset.
- **spaCy:** Used for Natural Language Processing (NLP) tasks, including Named Entity Recognition (NER) and sentiment analysis on Amazon product reviews.
- **Jupyter Notebook (VS Code Extension):** Used as the main environment for writing, running, and documenting code interactively.
- **Matplotlib:** Used for visualizing training results and performance graphs.

3.Model Summary

a. Iris Dataset (Scikit-learn)

A Decision Tree Classifier was trained to predict iris flower species based on petal and sepal dimensions.
The model achieved an accuracy of 1.00, showing perfect classification performance after preprocessing and label encoding.



b. MNIST Handwritten Digits (TensorFlow)

A CNN model was built to recognize handwritten digits.
After 5 training epochs, the model achieved 99.07% test accuracy, successfully visualizing predictions and accuracy graphs.
TensorFlow was chosen over PyTorch because the dataset was small and lightweight, and TensorFlow's Keras API is beginner-friendly and well-suited for visualizing metrics.

Deep Learning with TensorFlow – MNIST Handwritten Digit Classification

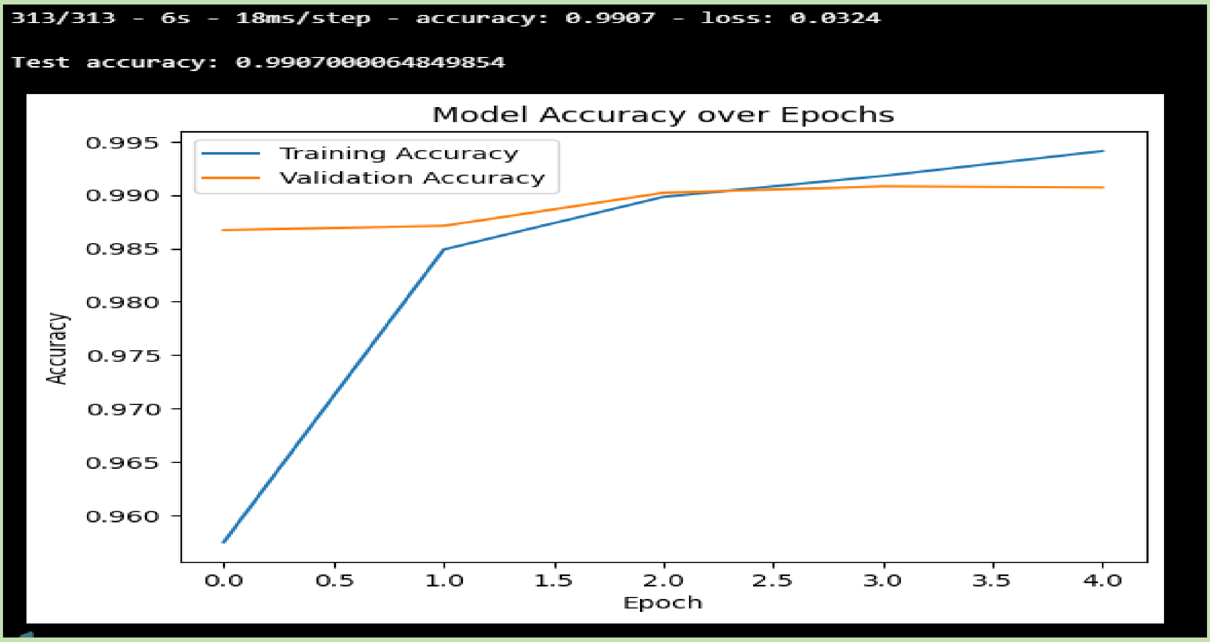
I used the MNIST dataset containing 70,000 grayscale images of handwritten digits (0–9). The data was normalized to improve model performance. A Convolutional Neural Network (CNN) with two convolutional and pooling layers was built using TensorFlow and Keras.

After training for 5 epochs, the model achieved:

- **Training Accuracy:** 99.4%
- **Validation Accuracy:** 99.1%
- **Loss:** 0.03

The CNN successfully learned to identify digits with high precision. Figure 2 shows the accuracy trend over epochs, and Figure 3 displays five sample predictions compared to true labels.

Figure 2: Training and Validation Accuracy Graph



```
313/313 - 6s - 18ms/step - accuracy: 0.9907 - loss: 0.0324

Test accuracy: 0.9907000064849854
```

c. Amazon Reviews (spaCy)

Using spaCy's pre-trained language model, Named Entity Recognition (NER) was performed to extract product names and brands.
A simple rule-based sentiment analysis categorized reviews as positive or negative, demonstrating spaCy's ability to process contextual language beyond basic string operations.

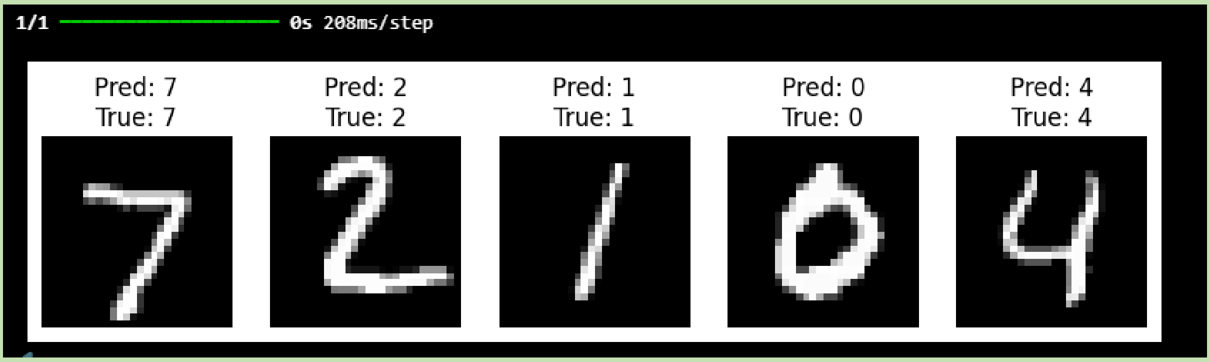
NLP with spaCy – Amazon Reviews Analysis

I used spaCy, a powerful NLP library, to analyze Amazon product reviews. Using the en_core_web_sm model, we performed Named Entity Recognition (NER) to identify entities such as product names and brands (e.g., "Samsung", "Apple MacBook", "Dell laptop").

Next, a simple rule-based sentiment analysis was implemented using keyword matching to classify reviews as *Positive*, *Negative*, or *Neutral*.

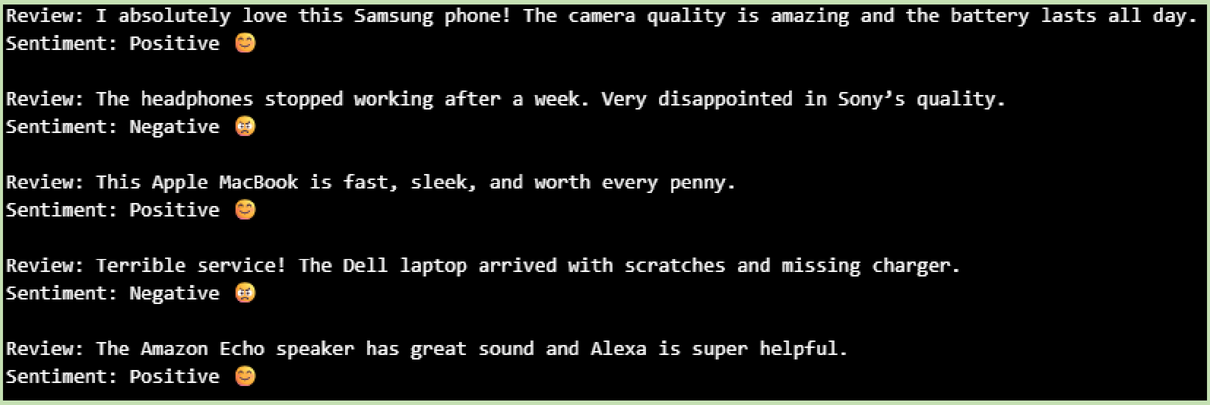
This demonstrates how spaCy simplifies text processing and entity extraction compared to basic Python string operations.

Figure 2: Extracted Named Entities from Amazon Reviews



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 208ms/step
```

Figure 5: Rule-Based Sentiment Classification Results



```
Review: I absolutely love this Samsung phone! The camera quality is amazing and the battery lasts all day.
Sentiment: Positive 😊

Review: The headphones stopped working after a week. Very disappointed in Sony’s quality.
Sentiment: Negative 😠

Review: This Apple MacBook is fast, sleek, and worth every penny.
Sentiment: Positive 😊

Review: Terrible service! The Dell laptop arrived with scratches and missing charger.
Sentiment: Negative 😠

Review: The Amazon Echo speaker has great sound and Alexa is super helpful.
Sentiment: Positive 😊
```

4. Results and Discussion

Each model performed effectively for its respective task:

- The **Scikit-learn** model provided fast and accurate predictions for structured data.
- The **TensorFlow CNN** achieved high accuracy with minimal overfitting, proving effective for image recognition.
- The **spaCy** model accurately identified named entities and sentiment, showing how NLP frameworks enhance text understanding.

These results demonstrate the versatility of AI tools from structured data analysis to visual and textual intelligence.

5.Theoretical questions

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

TensorFlow is a high-level, production-focused framework that simplifies model building through Keras and supports easy deployment on mobile and web

platforms. PyTorch, on the other hand, is more flexible and research-oriented, giving developers full control over model training and debugging.
You'd choose TensorFlow for quick prototyping or deploying scalable models, and PyTorch for research, experimentation, or custom model architecture.
I chose TensorFlow over PyTorch because my dataset was small and lightweight, and I only needed it for deep learning tasks such as building and visualizing model performance graphs and metrics. Additionally, as a beginner, I found TensorFlow (especially with Keras) easier to use and more beginner-friendly.

Describe two use cases for Jupyter Notebooks in AI development.

1. Developers can write, test, and visualize code step-by-step, making it ideal for exploring datasets and tuning models.
2. Notebooks combine code, outputs, and explanations in one place, which helps in creating reproducible AI reports and tutorials.

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

SpaCy provides pre-trained language models and linguistic features such as tokenization, named entity recognition, and part-of-speech tagging — all optimized for speed and accuracy.
Unlike basic Python string operations that only handle text patterns, SpaCy understands context and structure, making it powerful for advanced NLP tasks like sentiment analysis or entity extraction.

Comparative Analysis: Scikit-learn vs TensorFlow

Feature	Scikit-learn	TensorFlow
Target Applications	Classical machine learning (e.g., regression, classification, clustering)	Deep learning (e.g., neural networks, CNNs, RNNs)
Ease of Use for Beginners	Easier — simple syntax and consistent API	Moderate — more complex, especially for custom models
Community Support	Strong academic and ML community	Very large community with enterprise and research backing (Google)

1. Ethical Considerations

Even though the **MNIST** dataset is generally unbiased since it consists of handwritten digits, biases can still occur; for example, if the model performs better on digits written in certain handwriting styles like those from specific age groups or

cultures.
For the Amazon Reviews dataset, bias can arise from imbalanced sentiment: for instance, if there are more positive than negative reviews, the model might learn to classify most texts as positive.

Mitigation:

- TensorFlow Fairness Indicators can be used to measure and visualize model fairness across subgroups, helping identify where the model might be performing unfairly.
- spaCy's rule-based systems can help ensure balanced handling of language by controlling how entities and sentiment rules are applied, reducing bias caused by word choice or phrasing.

2. Troubleshooting Challenge

Common TensorFlow errors include dimension mismatches, wrong loss functions, and incorrect activations.
These can be fixed by reshaping inputs properly, using suitable loss functions (e.g., *categorical crossentropy* for classification), and ensuring correct output activations like *softmax*. This improves model accuracy and stability.

5. Conclusion

This assignment showcased how diverse AI tools can be combined to solve complex problems.
Through Scikit-learn, TensorFlow, and spaCy, I learned the importance of selecting the right framework for the right task — balancing performance, ease of use, and ethical fairness.
The project improved my understanding of both classical machine learning and deep learning, as well as responsible AI development practices.