Contents

Introduction

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

Student: Peter Kinyanjui Munga

Q1: Primary Differences between TensorFlow and PyTorch

Key Differences	TensorFlow	PyTorch
Computation Graphs	Uses static computation graphs, defined before execution; optimizes performance but less flexible.	Utilizes dynamic computation graphs, allowing flexibility and ease of debugging; built on-the-fly.
Ease of Use	Initially steep learning curve; now more user-friendly with TensorFlow 2.0 and Keras API.	Generally considered more intuitive and easier to use, especially for researchers and prototyping.
Ecosystem and Deployment	Extensive ecosystem including TensorFlow Serving, TensorFlow Lite, and TensorFlow.js.	Developing ecosystem with TorchScript and PyTorch Mobile, but still behind TensorFlow.

When to Choose One Over the Other:

Choose TensorFlow when:	Choose PyTorch when:
You need to deploy models in production environments.	You are engaged in research or prototyping.
You require support for mobile or web applications.	You need flexibility in model building and debugging.
You prefer a more extensive ecosystem for model serving and scaling.	You prefer a more Pythonic and user-friendly interface.

Q2: Use Cases for Jupyter Notebooks in AI Development

1. Interactive Data Exploration:
    a. Jupyter Notebooks allow data scientists to explore datasets interactively. They can visualize data distributions, run exploratory data analysis (EDA), and generate plots using libraries like Matplotlib and Seaborn. This iterative process helps in understanding the data better and making informed decisions for model development.
2. Documentation and Sharing:
    a. Jupyter Notebooks can combine code, visualizations, and narrative text, making them ideal for documenting the AI development process. This is particularly useful for sharing findings with stakeholders or collaborators, as notebooks can be easily converted to HTML or PDF formats for presentations.

Q3: spaCy vs. Basic Python String Operations in NLP

Feature	SPAcy	Basic Python
Tokenization	Provides advanced tokenization that recognizes words, punctuation, and special characters, handling various languages and linguistic nuances effectively.	Simple string operations like split() do not consider linguistic rules, leading to inaccuracies in tokenization, especially with contractions and punctuation.
Named Entity Recognition (NER)	Offers built-in NER capabilities, allowing for the identification and classification of entities (like names, dates, and organizations) in text with high accuracy.	Requires manual implementation of logic for entity recognition, which is often inefficient and less accurate compared to spaCy's pre-trained models.

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.

Criteria	Scikit-learn	TensorFlow
Target Applications	- Primarily designed for classical machine learning tasks.	- Designed for deep learning and neural network development.
	- Supports algorithms like SVM, decision trees, random forests, clustering, and dimensionality reduction.	- Supports complex architectures like CNNs, RNNs, and reinforcement learning.
	- Best suited for small to medium-sized datasets and traditional ML problems.	- Ideal for large datasets and high-dimensional data typical in deep learning applications.
Ease of Use for Beginners	- Generally easier for beginners to grasp due to its simple and consistent API.	- Steeper learning curve, especially in earlier versions (prior to TensorFlow 2.0).
	- Extensive documentation and tutorials make it accessible for those new to machine learning.	- While TensorFlow 2.0 introduced Keras for a more user-friendly interface, beginners may still find it complex compared to Scikit-learn.
Community Support	- Strong community support with a wealth of resources, including documentation, tutorials, and forums.	- Very large and active community, especially in deep learning.
	- Widely used in academia and industry for classical ML, leading to a large number of community-contributed resources.	- Extensive documentation, forums, and numerous tutorials available.
		- Strong backing from Google, which contributes to its ongoing development and support.

Ethical Considerations

Identifying and mitigating biases in machine learning models is crucial for ensuring fairness and accuracy. Here’s a breakdown of potential biases in the MNIST and Amazon Reviews models, along with how tools like TensorFlow Fairness Indicators and spaCy’s rule-based systems can help mitigate these biases.

Potential Biases

1. MNIST Model Biases

- Class Imbalance: The MNIST dataset contains images of handwritten digits (0-9), and although it is relatively balanced, slight imbalances can still lead to biases in prediction accuracy for certain digits (e.g., misclassifying '1' or '7' more often).

- Variability in Handwriting: The model may perform poorly on digits that are less frequently represented in the training set or on digits written in styles that differ significantly from those in the training set.

- Overfitting: If the model is too complex, it may overfit to the training data and not generalize well to unseen data, particularly if that data includes digits written in different styles.

2. Amazon Reviews Model Biases

- Sentiment Bias: The rule-based sentiment analysis may not capture the nuances of language, leading to misclassification of sentiment (e.g., sarcasm or mixed sentiments).

- Entity Recognition Bias: The NER model may not recognize less common products or brands, leading to incomplete extraction of relevant entities.

- Cultural Bias: Reviews from different demographics or regions may be interpreted differently, leading to biased sentiment analysis based on the language used.

Mitigation Strategies

TensorFlow Fairness Indicators

TensorFlow Fairness Indicators is a tool that helps evaluate and visualize the fairness of machine learning models. Here’s how it can help:

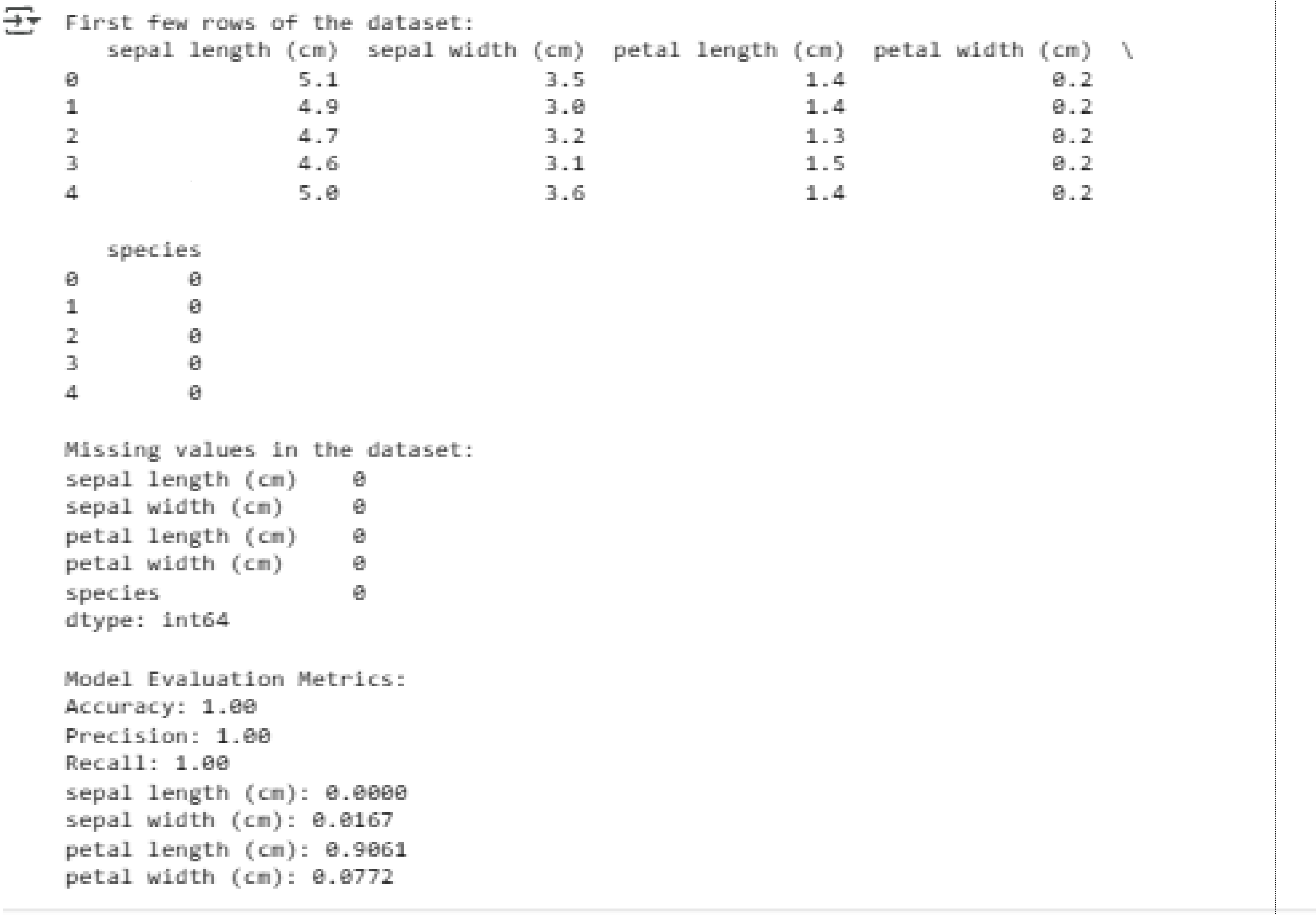
- Bias Detection: It provides metrics to evaluate model performance across different subgroups (e.g., different digit classes in MNIST or different product categories in Amazon Reviews). This helps identify any disparities in accuracy, precision, or recall.

- Visualization: The tool allows users to visualize fairness metrics, making it easier to spot biases that might not be evident from numerical results alone.

- Thresholding and Calibration: It can help in adjusting decision thresholds for different groups to achieve fairness without significantly sacrificing overall accuracy.

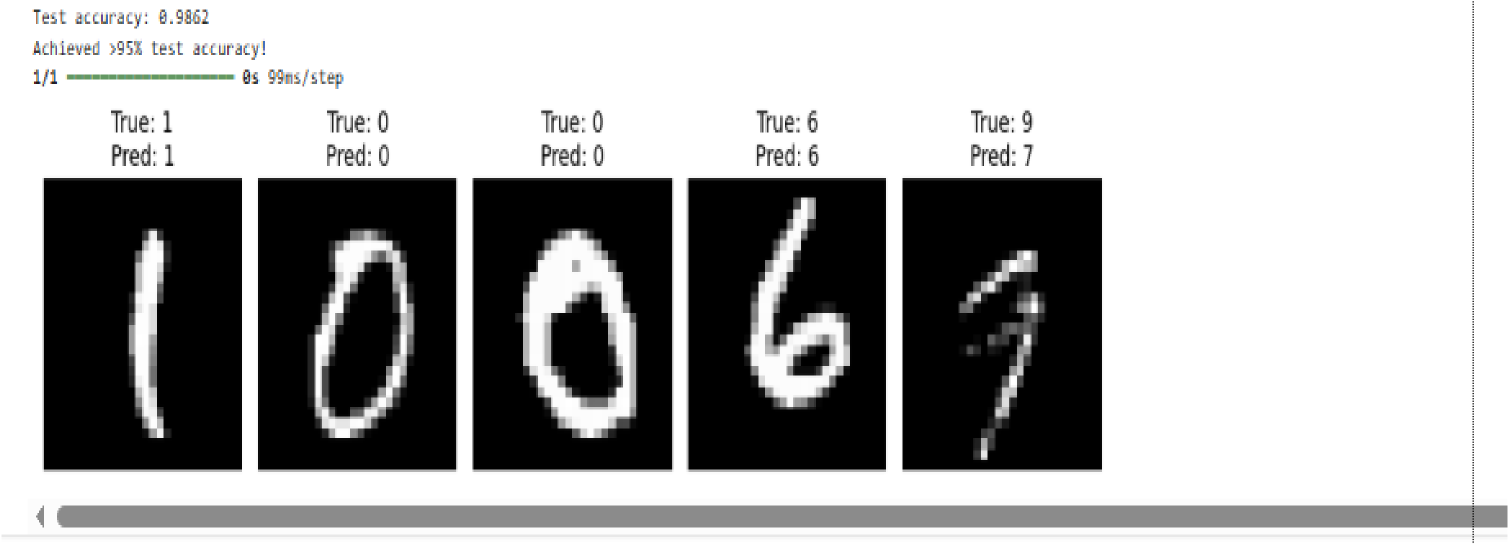
Screenshots of model accuracies:

Task 1: Classical ML with Scikit-learn



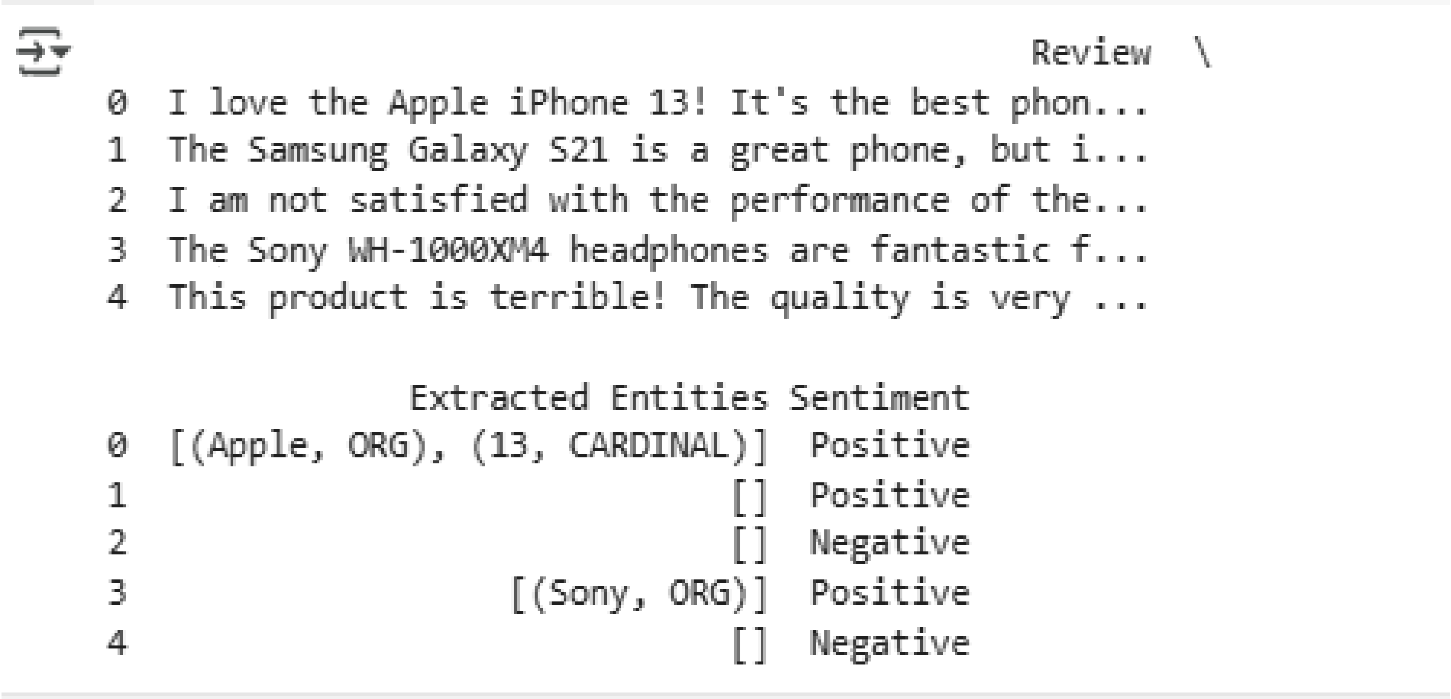
```
First few rows of the dataset:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                5.1               3.5                1.4               0.2
1                4.9               3.0                1.4               0.2
2                4.7               3.2                1.3               0.2
3                4.6               3.1                1.5               0.2
4                5.0               3.6                1.4               0.2

   species
0        0
1        0
2        0
3        0
4        0

Missing values in the dataset:
sepal length (cm)    0
sepal width (cm)     0
petal length (cm)    0
petal width (cm)     0
species              0
dtype: int64

Model Evaluation Metrics:
Accuracy: 1.00
Precision: 1.00
Recall: 1.00
sepal length (cm): 0.0000
sepal width (cm): 0.0167
petal length (cm): 0.9061
petal width (cm): 0.0772
```

Task 2: Deep Learning with TensorFlow/PyTorch



```
Test accuracy: 0.9862
Achieved >95% test accuracy!
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 99ms/step
```

Task 3: NLP with spaCy



```
                                               Review  \
0  I love the Apple iPhone 13! It's the best phon...
1  The Samsung Galaxy S21 is a great phone, but i...
2  I am not satisfied with the performance of the...
3  The Sony WH-1000XM4 headphones are fantastic f...
4  This product is terrible! The quality is very ...

             Extracted Entities Sentiment
0  [(Apple, ORG), (13, CARDINAL)]  Positive
1                              []  Positive
2                              []  Negative
3                   [(Sony, ORG)]  Positive
4                              []  Negative
```