Comparative Analysis of Scikit-learn and TensorFlow Models

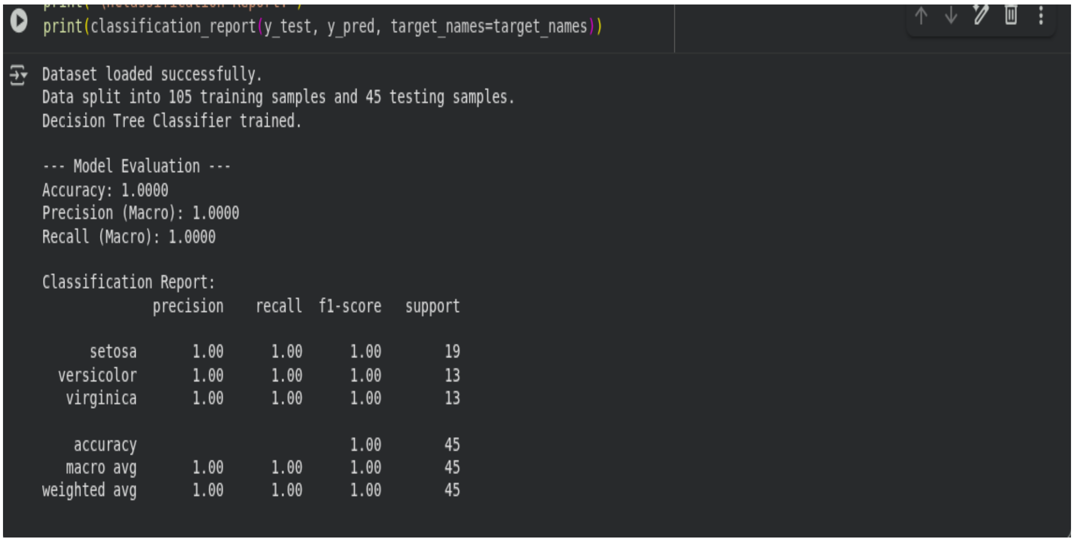
Prepared by Michelle

Part 2: Comparative Analysis

Feature	Scikit-learn	TensorFlow
Target Applications	Classical Machine Learning. Best for regression, classification, [illegible]	Deep Learning and Large-Scale ML. Best for [illegible]
Ease of Use for Beginners	Extremely high. Consistent API (.fit(), .predict(), .transform()). [illegible]	Moderate to high. Easier with Keras, but [illegible]
Community Support	Vast and mature. Huge Stack Overflow presence, strong documentation [illegible]	Massive and active. Backed by Google, with extensive [illegible]

Model Outputs and Performance

Figure 1: Scikit-learn Decision Tree Classifier Results



```
print(classification_report(y_test, y_pred, target_names=target_names))

Dataset loaded successfully.
Data split into 105 training samples and 45 testing samples.
Decision Tree Classifier trained.

--- Model Evaluation ---
Accuracy: 1.0000
Precision (Macro): 1.0000
Recall (Macro): 1.0000

Classification Report:
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        19
  versicolor       1.00      1.00      1.00        13
   virginica       1.00      1.00      1.00        13

    accuracy                           1.00        45
   macro avg       1.00      1.00      1.00        45
weighted avg       1.00      1.00      1.00        45
```

Figure 2: TensorFlow CNN Model Summary

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
Training data shape: (60000, 28, 28, 1)
Test data shape: (10000, 28, 28, 1)
/usr/local/lib/python3.12/dist-packages/keras/src/layers/convolutional/base_conv.py:113: UserWarning: Do not pass an `input_shape
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Model: "sequential"
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_1 (Conv2D)	(None, 11, 11, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 5, 5, 64)	0
flatten (Flatten)	(None, 1600)	0
dense (Dense)	(None, 128)	204,928
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 10)	1,290

```
Total params: 225,034 (879.04 KB)
Trainable params: 225,034 (879.04 KB)
Non-trainable params: 0 (0.00 B)
```

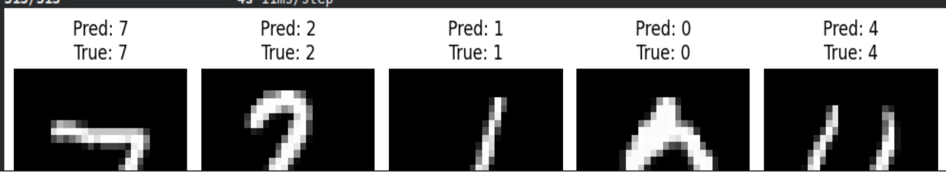
Figure 3: TensorFlow CNN Model Training and Predictions

```
--- Training Model ---
Epoch 1/5
422/422 ━━━━━━━━━━━━━━━━━━━━ 45s 102ms/step - accuracy: 0.7952 - loss: 0.6533 - val_accuracy: 0.9783 - val_loss: 0.0659
Epoch 2/5
422/422 ━━━━━━━━━━━━━━━━━━━━ 83s 106ms/step - accuracy: 0.9673 - loss: 0.1116 - val_accuracy: 0.9880 - val_loss: 0.0444
Epoch 3/5
422/422 ━━━━━━━━━━━━━━━━━━━━ 80s 101ms/step - accuracy: 0.9778 - loss: 0.0744 - val_accuracy: 0.9883 - val_loss: 0.0385
Epoch 4/5
422/422 ━━━━━━━━━━━━━━━━━━━━ 82s 102ms/step - accuracy: 0.9816 - loss: 0.0623 - val_accuracy: 0.9882 - val_loss: 0.0449
Epoch 5/5
422/422 ━━━━━━━━━━━━━━━━━━━━ 90s 121ms/step - accuracy: 0.9852 - loss: 0.0509 - val_accuracy: 0.9908 - val_loss: 0.0337

--- Evaluating Model ---
313/313 - 4s - 14ms/step - accuracy: 0.9906 - loss: 0.0269

Test accuracy: 0.9906

--- Visualizing Predictions ---
313/313 ━━━━━━━━━━━━━━━━━━━━ 4s 11ms/step
```



Part 3: Ethics & Optimization

Developing AI models carries significant ethical responsibilities, particularly concerning bias. **Bias in Datasets:** Both the MNIST and Amazon Reviews datasets can contain hidden biases. MNIST may not represent the full diversity of human handwriting styles, potentially performing worse for underrepresented groups. The Amazon reviews dataset could be skewed by demographic, cultural, or product-category biases, leading the NLP model to misunderstand sentiment from different user groups. **Mitigation Strategies:** Tools like TensorFlow Fairness Indicators can be used to audit model performance across different data slices, helping to identify and address performance gaps. For the NLP task, spaCy's rule-based systems offer transparency. By manually curating and expanding sentiment keyword lists and linguistic rules, we can create a system that is less of a "black box" and can be more easily audited for fairness and corrected for biases. It is crucial to be proactive in identifying potential sources of bias and leveraging tools and methodologies to build more equitable models.