Ethical Considerations

Bias in MNIST:
The MNIST dataset may not represent diverse handwriting styles. This can cause unfair predictions for users outside the dataset's demographic. Using TensorFlow Fairness Indicators allows us to detect and visualize bias across subgroups, ensuring fairness and equal accuracy.

Bias in Amazon Reviews Model:
The sentiment analysis uses a limited rule-based system that might misinterpret sarcasm or cultural language differences. Expanding the lexicon, handling negations, or combining spaCy with a trained model can reduce these errors.

Mitigation Tools:

- *TensorFlow Fairness Indicators* for fairness metrics and bias detection.
- *spaCy's rule-based pipelines* for improving entity and sentiment accuracy through better linguistic rules.

---

2 Troubleshooting Challenge

You'll be given (or can simulate) a **buggy TensorFlow model** — the goal is to fix it.
If you haven't been provided one, we can simulate a typical issue for demonstration.

Example Buggy Code

Here's a common example with a **dimension mismatch** in TensorFlow:

```
import tensorflow as tf
from tensorflow.keras import layers, models

# Buggy: Incorrect input shape
model = models.Sequential([
    layers.Dense(128, activation='relu', input_shape=(28, 28, 1)),  # ❌ should be flattened
    layers.Dense(10, activation='softmax')
])

# Buggy: Wrong loss function for integer labels
model.compile(optimizer='adam', loss='mse', metrics=['accuracy'])
```

Fixed Version

```
import tensorflow as tf
from tensorflow.keras import layers, models

# Correct input: flatten before Dense
model = models.Sequential([
    layers.Flatten(input_shape=(28, 28)),
    layers.Dense(128, activation='relu'),
    layers.Dense(10, activation='softmax')
])

# Correct loss for classification
model.compile(optimizer='adam',
loss='sparse_categorical_crossentropy', metrics=['accuracy'])

print(model.summary())
```

Explanation of Fixes:

1. **Flatten Layer Added:** Converts 28×28 images into 784-length vectors.
2. **Loss Function Fixed:** Changed from `mse` → `sparse_categorical_crossentropy` (correct for multi-class classification).
3. **Metrics Retained:** Accuracy is appropriate for classification.

Report Section (Example Text)

Troubleshooting Challenge:

The provided TensorFlow code produced a dimension mismatch error due to an incorrect input shape and loss function.
To fix it:

- A `Flatten()` layer was added to reshape image inputs.

- The loss function was corrected from `mse` to `sparse_categorical_crossentropy`.

After correction, the model trained successfully and achieved the expected accuracy.