Part 3: Ethics & Optimization (10%)

1. Ethical Considerations

Potential Biases in MNIST or Amazon Reviews Models:
MNIST Model (Handwritten Digit Recognition): The MNIST dataset mainly consists of digits written by individuals from certain demographic or cultural backgrounds (e.g., specific handwriting styles common in Western education systems). This may cause poor performance on digits written by people with different writing patterns, ages, or handwriting conditions.
Amazon Reviews Sentiment Model: The dataset may reflect biased human opinions—such as gender, cultural, or product-category bias. For example, words used in negative reviews for certain products might carry gender or cultural stereotypes, leading to unfair misclassification.

Bias Mitigation Tools and Strategies:
TensorFlow Fairness Indicators: Evaluates model performance across different slices of data (e.g., gender, age, or language group) and provides fairness metrics like false positive rate and false negative rate. This helps detect demographic imbalances in performance.
spaCy's Rule-Based Systems: Enables custom text-processing rules to detect or neutralize biased language before training or inference. For example, rule-based preprocessing can flag gendered or culturally biased words and ensure context-aware correction.

2. Troubleshooting Challenge

Example Buggy Code (Before Debugging):
model = tf.keras.Sequential([tf.keras.layers.Dense(128, input_shape=(28, 28), activation='relu'), tf.keras.layers.Dense(10, activation='softmax')]) model.compile(optimizer='adam', loss='binary_crossentropy', # Wrong loss for multi-class metrics=['accuracy']) model.fit(x_train, y_train, epochs=5)

Fixed Code (After Debugging):
model = tf.keras.Sequential([tf.keras.layers.Flatten(input_shape=(28, 28)), # Flatten input properly tf.keras.layers.Dense(128, activation='relu'), tf.keras.layers.Dense(10, activation='softmax')]) model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', # Correct loss for integer labels metrics=['accuracy']) model.fit(x_train, y_train, epochs=5)
Fix Summary: Added a Flatten layer to reshape input data from 2D (28×28) to 1D (784), changed the loss function to sparse_categorical_crossentropy, and ensured the output layer uses softmax activation for multi-class classification.

Bonus Task (Extra 10%)

Deploy Your Model (Using Streamlit or Flask):

import streamlit as st import tensorflow as tf from PIL import Image import numpy as np st.title("MNIST Digit Classifier") uploaded_file = st.file_uploader("Upload a digit image", type=["png", "jpg"]) if uploaded_file is not None: image = Image.open(uploaded_file).convert('L').resize((28, 28)) img_array = np.expand_dims(np.array(image) / 255.0, axis=(0, -1)) model = tf.keras.models.load_model("mnist_model.h5") prediction = np.argmax(model.predict(img_array)) st.success(f"Predicted Digit: {prediction}")
Output: A simple Streamlit web interface that allows users to upload an image of a handwritten digit. The model predicts the digit and displays it instantly. This can be hosted on Streamlit Cloud or Flask and submitted with a screenshot and live link.