AI Assignment - Part 3: Ethics & Optimization

# 1. Ethical Considerations

Potential Biases:  
  
- MNIST Model (CNN):  
 - The dataset is balanced across digits 0–9, but real-world digit images may have different fonts, rotations, or cultural symbols not present in MNIST.  
 - Bias risk: The model may perform poorly on handwriting styles from underrepresented groups (e.g., left-handed people, people from different age groups or regions).  
  
- Amazon Reviews Sentiment Model:  
 - Rule-based systems use static positive/negative word lists.  
 - Bias risk: These lists may not capture sarcasm, cultural context, or linguistic diversity (e.g., “That product is sick!” could mean good or bad).  
  
Mitigation Strategies:  
  
- TensorFlow Fairness Indicators:  
 - Can measure performance (accuracy, precision, recall) across subgroups (e.g., age, gender, region).  
 - Helps identify if your model unfairly favors or disadvantages a certain group.  
 - Apply it by adding metadata to your dataset and running subgroup analysis.  
  
- spaCy Rule-based Systems:  
 - Can be expanded to include more diverse slang, idioms, or multilingual support.  
 - You can fine-tune entity recognition patterns or define new ones using EntityRuler to reduce misclassification.

# 2. Troubleshooting Challenge

Buggy TensorFlow Code:

import tensorflow as tf  
  
model = tf.keras.Sequential([  
 tf.keras.layers.Dense(64, input\_shape=(28, 28), activation='relu'),  
 tf.keras.layers.Dense(10, activation='softmax')  
])  
  
model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])  
  
(X\_train, y\_train), (X\_test, y\_test) = tf.keras.datasets.mnist.load\_data()  
  
model.fit(X\_train, y\_train, epochs=5)

Problems Identified:

1. Input shape mismatch: Dense layers expect a flat vector, but input shape is (28, 28), not flattened.  
2. Wrong loss function: binary\_crossentropy is for binary classification, but MNIST has 10 classes.  
3. Missing normalization: Pixel values range 0–255; should scale to 0–1.  
4. Labels not encoded properly: sparse\_categorical\_crossentropy is better since labels are integers 0–9.

Fixed Code:

import tensorflow as tf  
  
# Load and preprocess data  
(X\_train, y\_train), (X\_test, y\_test) = tf.keras.datasets.mnist.load\_data()  
X\_train = X\_train.reshape(-1, 28 \* 28).astype('float32') / 255.0  
X\_test = X\_test.reshape(-1, 28 \* 28).astype('float32') / 255.0  
  
# Build a correct model  
model = tf.keras.Sequential([  
 tf.keras.layers.Dense(64, input\_shape=(784,), activation='relu'),  
 tf.keras.layers.Dense(10, activation='softmax')  
])  
  
# Correct loss function for multiclass classification  
model.compile(optimizer='adam',  
 loss='sparse\_categorical\_crossentropy',  
 metrics=['accuracy'])  
  
# Train the model  
model.fit(X\_train, y\_train, epochs=5)