Model Development Phase Template

Date	5 November 2024
Team ID	SWTID1726832093
Project Title	Analysis of Amazon Cell Phone Reviews Using NLP Technique
Maximum Marks	10 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include a summary and training and validation performance metrics for multiple models, presented through respective screenshots.

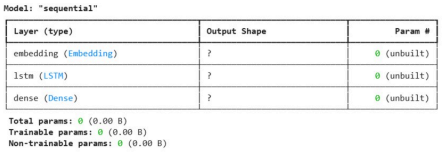
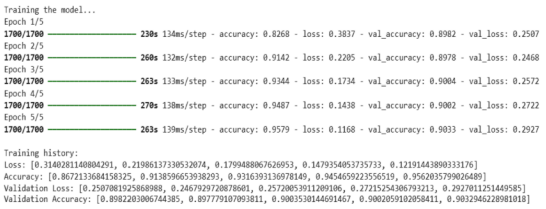
Initial Model Training Code (5 marks):

```python
from google.colab import files
import pandas as pd
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Embedding, LSTM
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from sklearn.model_selection import train_test_split

uploaded = files.upload()
for filename in uploaded.keys():
    print(f'Uploaded file: {filename}')
    data = pd.read_csv(filename)

print("\nDataFrame Preview:")
print(data.head())
X = data['body'].astype(str).values  # Reviews
y = data['rating'].apply(lambda x: 1 if x >= 4 else 0).values  # Binary sentiment: 1 for positive, 0 for negative

tokenizer = Tokenizer(num_words=10000)
tokenizer.fit_on_texts(X)
X_seq = tokenizer.texts_to_sequences(X)
X_pad = pad_sequences(X_seq, maxlen=100)
X_train, X_test, y_train, y_test = train_test_split(X_pad, y, test_size=0.2, random_state=42)
model = Sequential()
model.add(Embedding(input_dim=10000, output_dim=128, input_length=100))
model.add(LSTM(100))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
print("\nModel Summary:")
model.summary()
print("\nTraining the model...")
history = model.fit(X_train, y_train, epochs=5, batch_size=32, validation_data=(X_test, y_test))
print("\nTraining history:")
print("Loss:", history.history['loss'])
print("Accuracy:", history.history['accuracy'])
print("Validation Loss:", history.history['val_loss'])
print("Validation Accuracy:", history.history['val_accuracy'])
loss, accuracy = model.evaluate(X_test, y_test)
print(f"\nTest Loss: {loss}")
print(f"Test Accuracy: {accuracy}")
```

Model	Summary	Training and Validation Performance Metrics
Model 1		

Model Validation and Evaluation Report (5 marks):
