

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):

```
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"\nTest Accuracy: {accuracy}")
```

Model	Summary	Training and Validation Performance Metrics												
Model 1	<div>Model: "sequential"</div> <table><tr><th>Layer (type)</th><th>Output Shape</th><th>Param #</th></tr><tr><td>embedding (Embedding)</td><td>?</td><td>0 (unbuilt)</td></tr><tr><td>lstm (LSTM)</td><td>?</td><td>0 (unbuilt)</td></tr><tr><td>dense (Dense)</td><td>?</td><td>0 (unbuilt)</td></tr></table> <div>Total params: 0 (0.00 B) Trainable params: 0 (0.00 B) Non-trainable params: 0 (0.00 B)</div>	Layer (type)	Output Shape	Param #	embedding (Embedding)	?	0 (unbuilt)	lstm (LSTM)	?	0 (unbuilt)	dense (Dense)	?	0 (unbuilt)	<div>Training the model...</div> <div>Epoch 1/5 1700/1700 — 230s 134ms/step - accuracy: 0.8268 - loss: 0.3837 - val_accuracy: 0.8982 - val_loss: 0.2587</div> <div>Epoch 2/5 1700/1700 — 260s 132ms/step - accuracy: 0.9142 - loss: 0.2205 - val_accuracy: 0.8970 - val_loss: 0.2468</div> <div>Epoch 3/5 1700/1700 — 263s 133ms/step - accuracy: 0.9344 - loss: 0.1734 - val_accuracy: 0.9084 - val_loss: 0.2572</div> <div>Epoch 4/5 1700/1700 — 270s 138ms/step - accuracy: 0.9487 - loss: 0.1438 - val_accuracy: 0.9002 - val_loss: 0.2722</div> <div>Epoch 5/5 1700/1700 — 263s 139ms/step - accuracy: 0.9579 - loss: 0.1168 - val_accuracy: 0.9033 - val_loss: 0.2927</div> <div>Training history: Loss: [0.3140281148884291, 0.21086137130612074, 0.1794888067620953, 0.1479354053775713, 0.12191443898113176] Accuracy: [0.86721233684158325, 0.9138596653938293, 0.93163933136978149, 0.945465922356519, 0.9562835759826488] Validation Loss: [0.2587081925868988, 0.2467929720878681, 0.25720053911289106, 0.27215254306793213, 0.2927811251449585] Validation Accuracy: [0.8982203306744385, 0.897779107093811, 0.9003538144691467, 0.9002059102058411, 0.9032946228981018]</div>
Layer (type)	Output Shape	Param #												
embedding (Embedding)	?	0 (unbuilt)												
lstm (LSTM)	?	0 (unbuilt)												
dense (Dense)	?	0 (unbuilt)												

Model Validation and Evaluation Report (5 marks):
