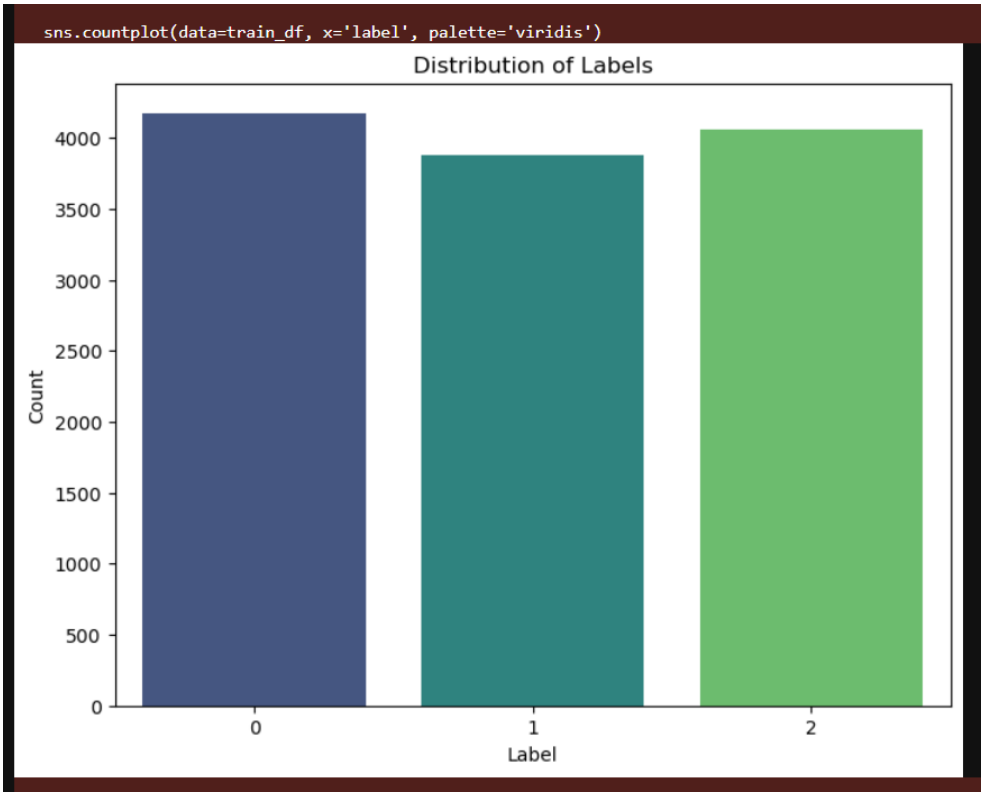
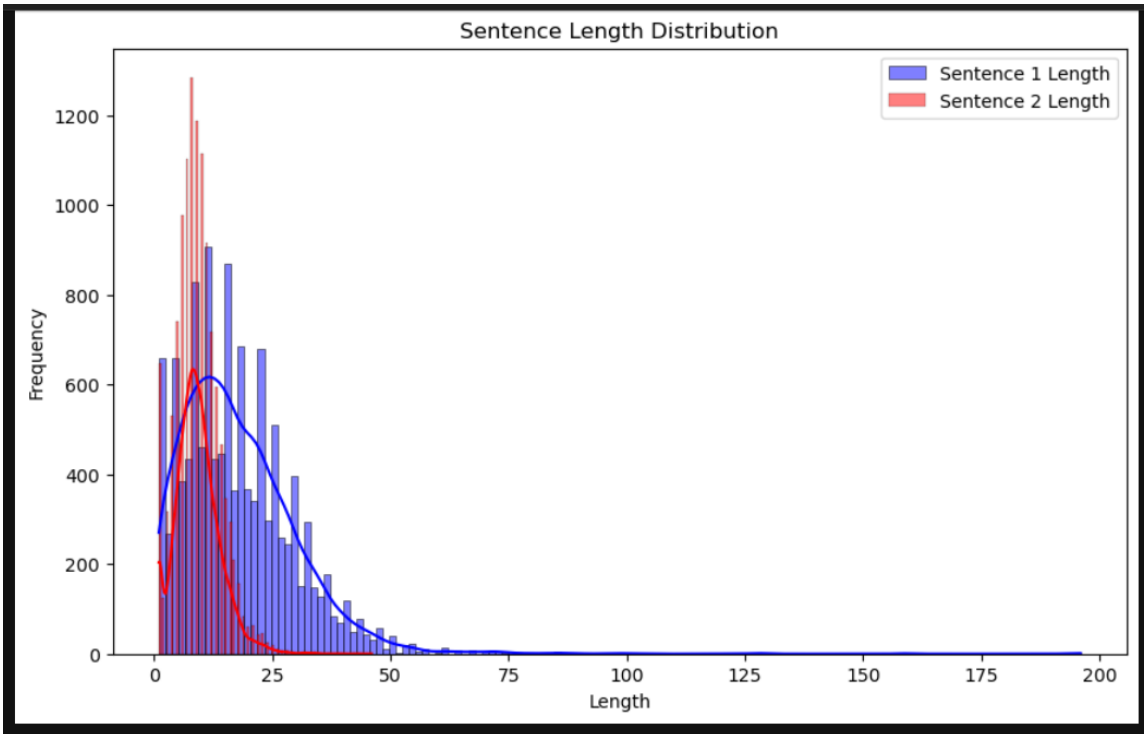
Step-1.EDA and Text Processing

```
sns.countplot(data=train_df, x='label', palette='viridis')
```





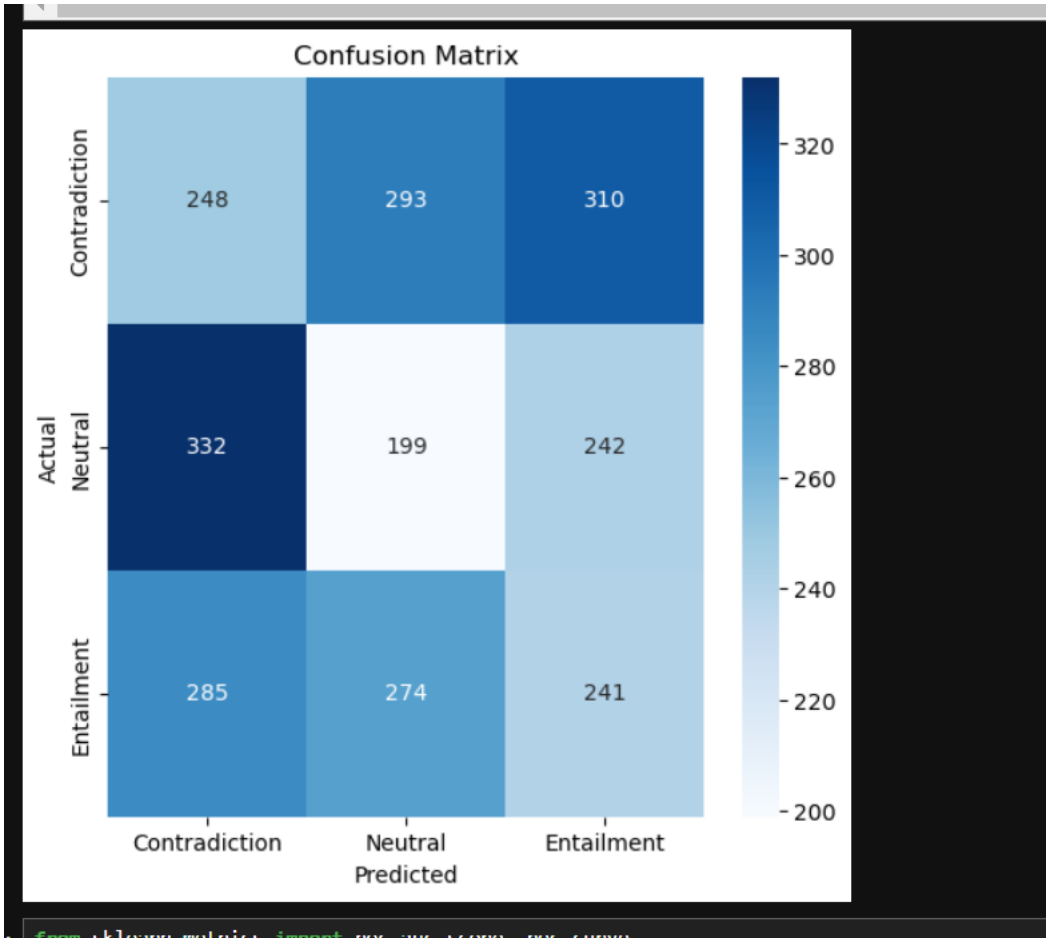

```
# Check for missing values
missing_values = train_df.isnull().sum()
print("Missing Values in the dataset:", missing_values)
```

Step-2. Regression Analysis

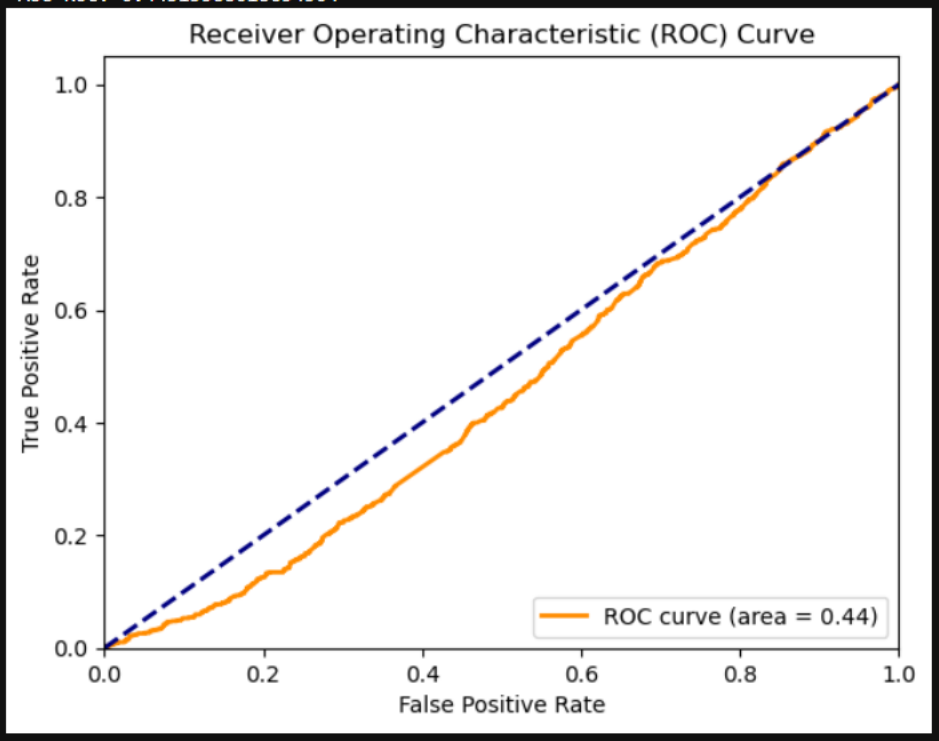
```
print("Random Forest Model Evaluation:")
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
Random Forest Model Evaluation:
Accuracy: 0.2838283828382838
Classification Report:
              precision    recall  f1-score   support

           0       0.29      0.29      0.29       851
           1       0.26      0.26      0.26       773
           2       0.30      0.30      0.30       800

    accuracy                           0.28      2424
   macro avg       0.28      0.28      0.28      2424
weighted avg       0.28      0.28      0.28      2424
```

```
# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
```

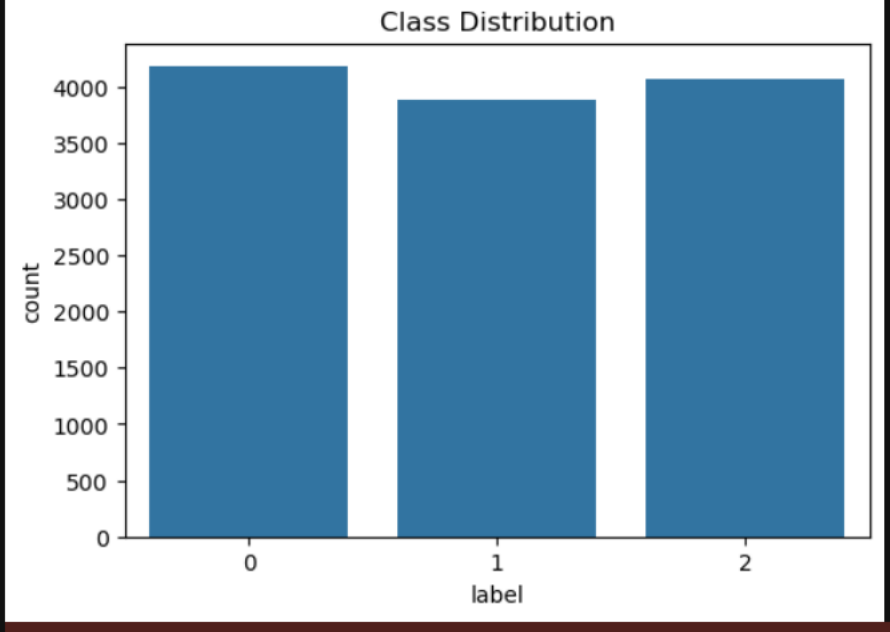


```
AUC-ROC: 0.44313588018654304
```




```
]: from sklearn.model_selection import GridSearchCV
```

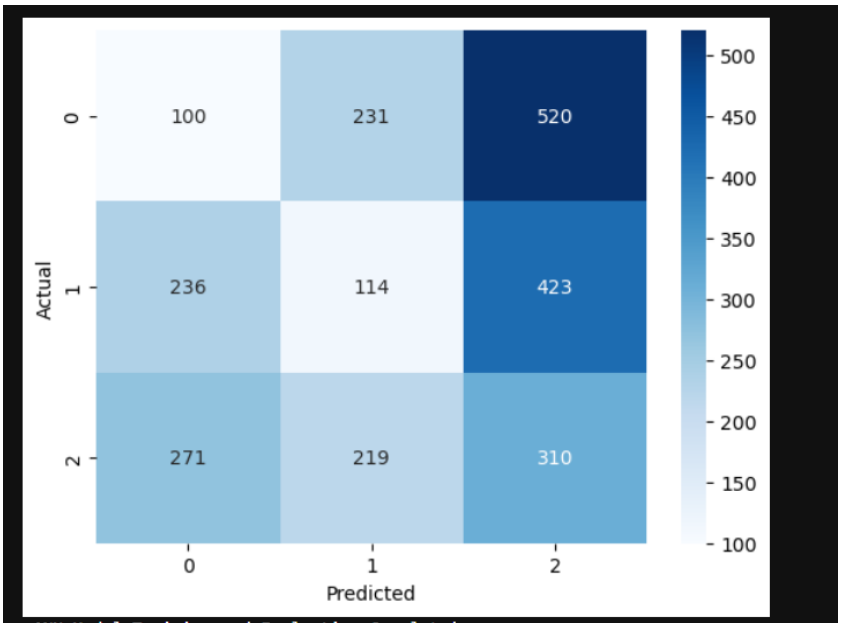
Step-3. ANN




```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Ayesha\AppData\Roaming\nltk_data...
```

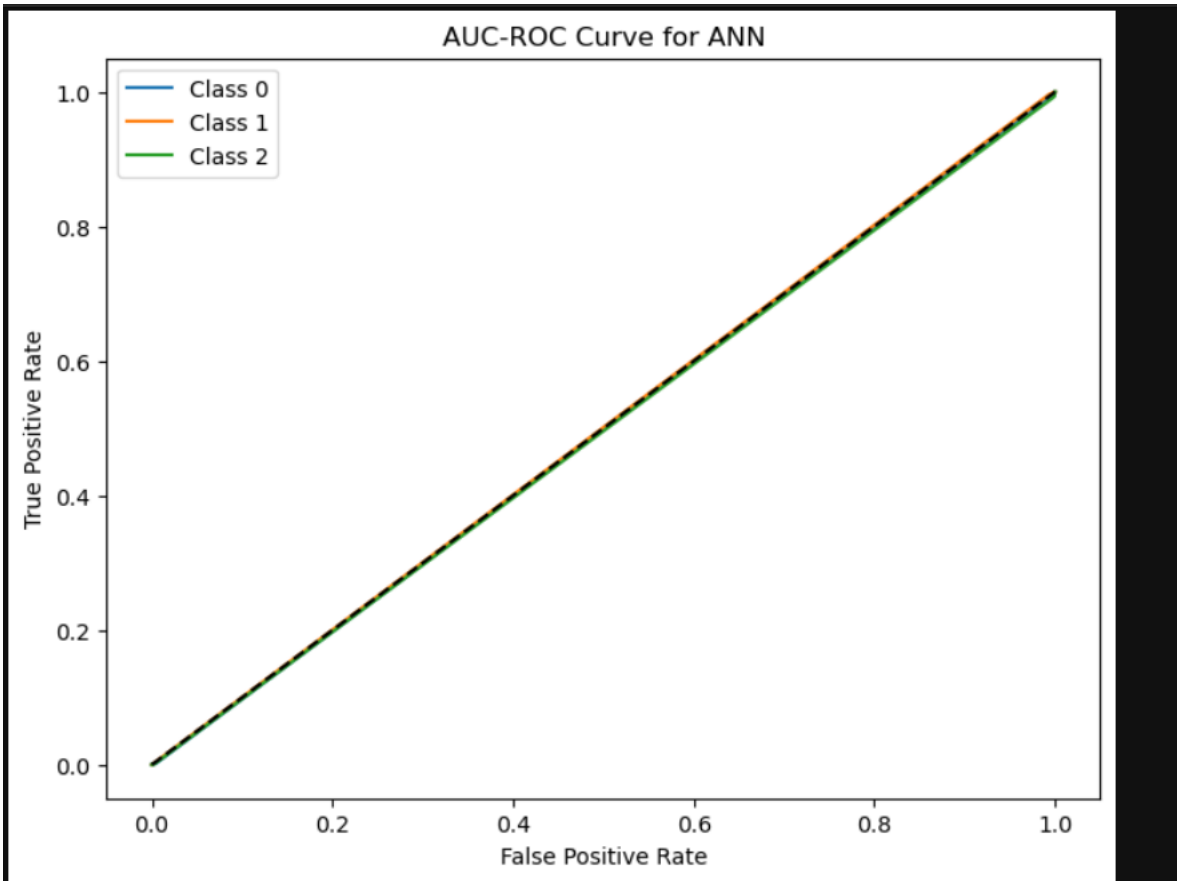
```
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Epoch 1/10
303/303 ━━━━━━━━━━━━━━━━━━━━ 37s 114ms/step - accuracy: 0.3252 - loss: 1.0999 - val_accuracy: 0.3568 - val_loss: 1.0985
Epoch 2/10
303/303 ━━━━━━━━━━━━━━━━━━━━ 34s 113ms/step - accuracy: 0.3834 - loss: 1.0896 - val_accuracy: 0.3474 - val_loss: 1.1002
Epoch 3/10
303/303 ━━━━━━━━━━━━━━━━━━━━ 35s 114ms/step - accuracy: 0.5053 - loss: 0.9741 - val_accuracy: 0.2364 - val_loss: 1.3246
Epoch 4/10
303/303 ━━━━━━━━━━━━━━━━━━━━ 34s 113ms/step - accuracy: 0.6320 - loss: 0.7387 - val_accuracy: 0.2273 - val_loss: 1.7801
Epoch 5/10
303/303 ━━━━━━━━━━━━━━━━━━━━ 34s 113ms/step - accuracy: 0.6909 - loss: 0.6076 - val_accuracy: 0.2199 - val_loss: 2.2458
Epoch 6/10
303/303 ━━━━━━━━━━━━━━━━━━━━ 34s 113ms/step - accuracy: 0.7170 - loss: 0.5281 - val_accuracy: 0.2294 - val_loss: 3.0246
Epoch 7/10
303/303 ━━━━━━━━━━━━━━━━━━━━ 34s 114ms/step - accuracy: 0.7447 - loss: 0.4719 - val_accuracy: 0.2244 - val_loss: 3.5608
Epoch 8/10
303/303 ━━━━━━━━━━━━━━━━━━━━ 34s 113ms/step - accuracy: 0.7463 - loss: 0.4444 - val_accuracy: 0.2219 - val_loss: 3.7406
Epoch 9/10
303/303 ━━━━━━━━━━━━━━━━━━━━ 34s 113ms/step - accuracy: 0.7874 - loss: 0.3908 - val_accuracy: 0.2224 - val_loss: 4.2665
Epoch 10/10
303/303 ━━━━━━━━━━━━━━━━━━━━ 35s 114ms/step - accuracy: 0.7960 - loss: 0.3864 - val_accuracy: 0.2162 - val_loss: 4.1295
76/76 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step
Accuracy: 0.21617161716171618
Classification Report:
              precision    recall  f1-score   support
```

```
76/76 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step
Accuracy: 0.21617161716171618
Classification Report:
              precision    recall  f1-score   support

           0       0.16      0.12      0.14       851
           1       0.20      0.15      0.17       773
           2       0.25      0.39      0.30       800

    accuracy                           0.22      2424
   macro avg       0.20      0.22      0.20      2424
weighted avg       0.20      0.22      0.20      2424
```



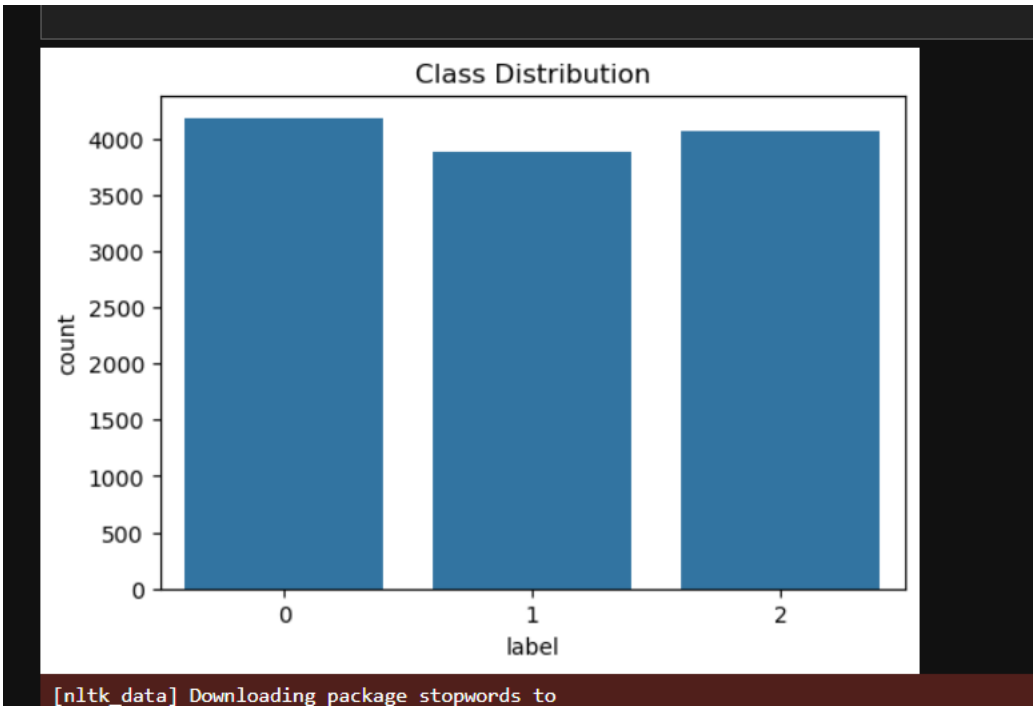

```
ANN Model Training and Evaluation Complete!
```

```
import pandas as pd
```



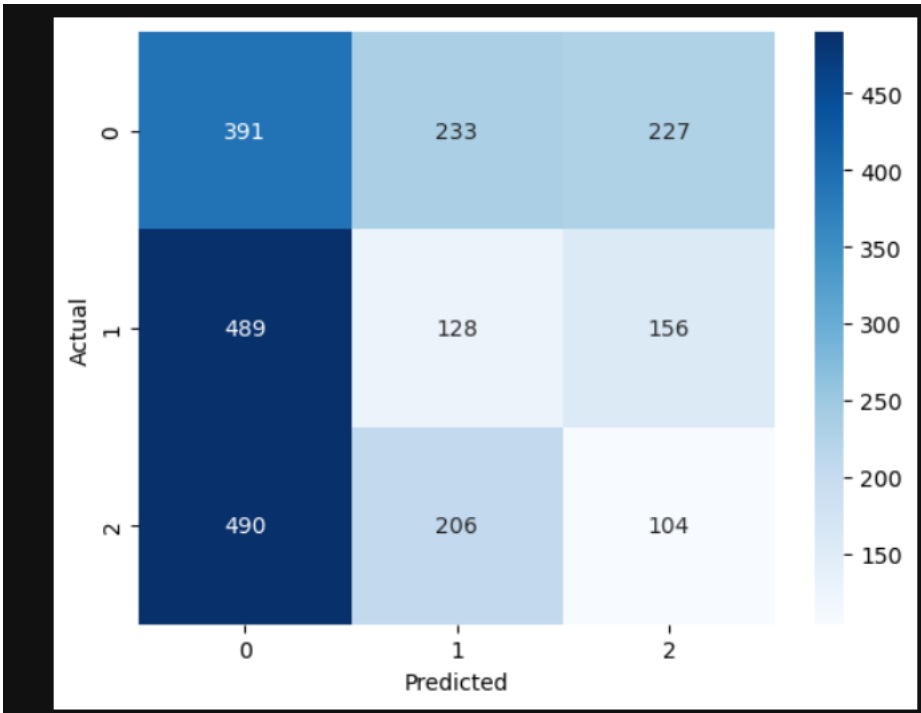

```
ANN Model Training and Evaluation Complete!
```

Step-4. GRU



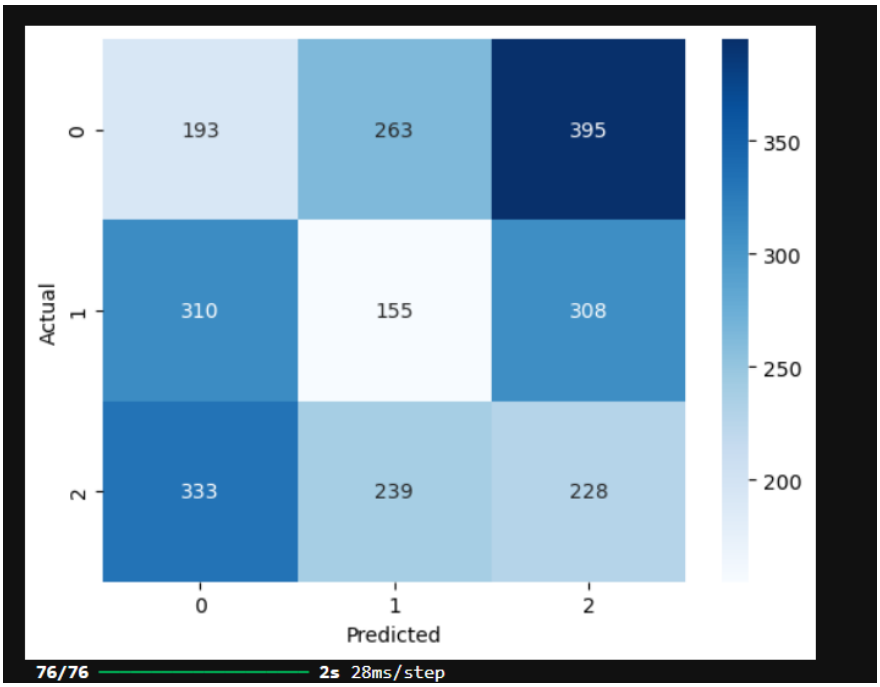

```
  warnings.warn(
303/303 ━━━━━━━━━━━━━━━━━━━━ 39s 106ms/step - accuracy: 0.3351 - loss: 1.1008 - val_accuracy: 0.3511 - val_loss: 1.0981
Epoch 2/10
303/303 ━━━━━━━━━━━━━━━━━━━━ 31s 102ms/step - accuracy: 0.3619 - loss: 1.0952 - val_accuracy: 0.3548 - val_loss: 1.0997
Epoch 3/10
303/303 ━━━━━━━━━━━━━━━━━━━━ 31s 104ms/step - accuracy: 0.4114 - loss: 1.0806 - val_accuracy: 0.2632 - val_loss: 1.2027
Epoch 4/10
303/303 ━━━━━━━━━━━━━━━━━━━━ 30s 100ms/step - accuracy: 0.5337 - loss: 0.9172 - val_accuracy: 0.2434 - val_loss: 1.4322
Epoch 5/10
303/303 ━━━━━━━━━━━━━━━━━━━━ 31s 103ms/step - accuracy: 0.6145 - loss: 0.7538 - val_accuracy: 0.2294 - val_loss: 1.8950
Epoch 6/10
303/303 ━━━━━━━━━━━━━━━━━━━━ 31s 102ms/step - accuracy: 0.6682 - loss: 0.6314 - val_accuracy: 0.2413 - val_loss: 2.5440
Epoch 7/10
303/303 ━━━━━━━━━━━━━━━━━━━━ 31s 102ms/step - accuracy: 0.7187 - loss: 0.5365 - val_accuracy: 0.2397 - val_loss: 3.2004
Epoch 8/10
303/303 ━━━━━━━━━━━━━━━━━━━━ 31s 101ms/step - accuracy: 0.7433 - loss: 0.4810 - val_accuracy: 0.2401 - val_loss: 3.6609
Epoch 9/10
303/303 ━━━━━━━━━━━━━━━━━━━━ 32s 105ms/step - accuracy: 0.7681 - loss: 0.4297 - val_accuracy: 0.2550 - val_loss: 4.3495
Epoch 10/10
303/303 ━━━━━━━━━━━━━━━━━━━━ 32s 105ms/step - accuracy: 0.7778 - loss: 0.4042 - val_accuracy: 0.2570 - val_loss: 4.3246
76/76 ━━━━━━━━━━━━━━━━━━━━ 4s 42ms/step
Accuracy: 0.257013201320132
Classification Report:
```

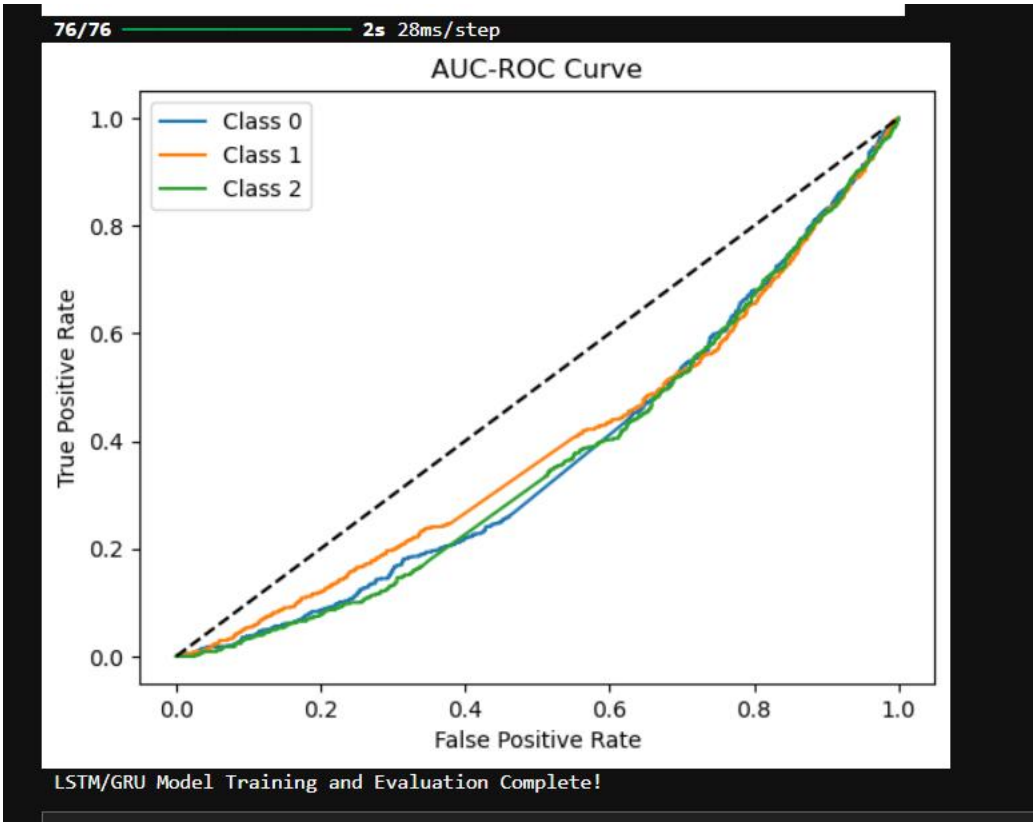
```
Accuracy: 0.257013201320132
Classification Report:
              precision    recall  f1-score   support

           0       0.29      0.46      0.35       851
           1       0.23      0.17      0.19       773
           2       0.21      0.13      0.16       800

    accuracy                           0.26      2424
   macro avg       0.24      0.25      0.23      2424
weighted avg       0.24      0.26      0.24      2424
```



LSTM/GRU Model Training and Evaluation Complete!

```
import pandas as pd
```


193
263
395
310
155
308
333
239
228
Actual
Predicted
0
1
2
350
300
250
200
76/76 2s 28ms/step


76/76 2s 28ms/step
AUC-ROC Curve
Class 0
Class 1
Class 2
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
LSTM/GRU Model Training and Evaluation Complete!

Step-5. BERT

```
All PyTorch model weights were used when initializing TFBertForSequenceClassification.

Some weights or buffers of the TF 2.0 model TFBertForSequenceClassification were not initialized from the PyTorch model and are newly initialized:
sifier.weight', 'classifier.bias']
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
WARNING:tensorflow:From C:\Users\Ayesha\AppData\Roaming\Python\Python312\site-packages\tf_keras\src\optimizers\__init__.py:317: The name tf.train.O
er is deprecated. Please use tf.compat.v1.train.Optimizer instead.

Epoch 1/3
WARNING:tensorflow:From C:\Users\Ayesha\AppData\Roaming\Python\Python312\site-packages\tf_keras\src\utils\tf_utils.py:492: The name tf.ragged.Ragge
rValue is deprecated. Please use tf.compat.v1.ragged.RaggedTensorValue instead.

WARNING:tensorflow:From C:\Users\Ayesha\AppData\Roaming\Python\Python312\site-packages\tf_keras\src\engine\base_layer_utils.py:384: The name tf.exe
_eagerly_outside_functions is deprecated. Please use tf.compat.v1.executing_eagerly_outside_functions instead.

 89/606 [===>.....................] - ETA: 1:15:03 - loss: 1.1984 - accuracy: 0.3174
```

```
import pandas as pd
```