```
#BiGRU Model For Manual Annotated dataset...............

import pandas as pd
import numpy as np
import re
from nltk.stem import WordNetLemmatizer
from sklearn.model_selection import KFold


from tensorflow.keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Embedding, Bidirectional, GRU, Dropout, Dense
from keras.optimizers import Adam
from sklearn.metrics import confusion_matrix, classification_report
import matplotlib.pyplot as plt
import seaborn as sns
from imblearn.over_sampling import RandomOverSampler

# 1. Load and Preprocess the dataset
df = pd.read_csv('R2_ChatGPt_dataset.csv', encoding='latin1')
lemmatizer = WordNetLemmatizer()

def clean_text(text):
    text = text.lower()
    text = re.sub(r'[^\w\s]', '', text)
    text = re.sub(r'\d+', '', text)
    tokens = text.split()
    tokens = [lemmatizer.lemmatize(token) for token in tokens]
    return ' '.join(tokens)

df['Base_Reviews'] = df['Base_Reviews'].apply(clean_text)

tokenizer = Tokenizer()
tokenizer.fit_on_texts(df['Base_Reviews'])
X = tokenizer.texts_to_sequences(df['Base_Reviews'])
vocab_size = len(tokenizer.word_index) + 1
maxlen = 100
X = pad_sequences(X, padding='post', maxlen=maxlen)

y_dict = {'anger': 0, 'confusion': 1, 'disappointment': 2, 'distrust':
3, 'disgust': 4, 'frustration': 5, 'fear': 6, 'sadness': 7}
y = df['Manual_annotation'].map(y_dict)
y = pd.get_dummies(df['Manual_annotation']).values

# 2. Oversample to balance classes
oversampler = RandomOverSampler(random_state=42)
X_resampled, y_resampled = oversampler.fit_resample(X, y)

# 3. Define the BiGRU model
```

```
def create_bigru_model():
    model = Sequential()
    model.add(Embedding(input_dim=vocab_size, output_dim=100,
input_length=maxlen))
    model.add(Bidirectional(GRU(64, return_sequences=True)))
    model.add(Dropout(0.2))
    model.add(Bidirectional(GRU(64)))
    model.add(Dropout(0.2))
    model.add(Dense(8, activation='softmax'))
    model.compile(optimizer=Adam(learning_rate=0.001),
loss='categorical_crossentropy', metrics=['accuracy'])
    return model

# 4. K-Fold Cross-Validation
n_folds = 10
kfold = KFold(n_splits=n_folds, shuffle=True)
fold_no = 1
acc_per_fold = []

# Lists to store average accuracies across folds
avg_train_acc = []
avg_val_acc = []

for train, test in kfold.split(X_resampled, y_resampled):
    model = create_bigru_model()
    history = model.fit(X_resampled[train], y_resampled[train],
validation_split=0.1, epochs=10, batch_size=32, verbose=0)

    avg_train_acc.append(history.history['accuracy'])
    avg_val_acc.append(history.history['val_accuracy'])

    scores = model.evaluate(X_resampled[test], y_resampled[test],
verbose=0)
    acc_per_fold.append(scores[1] * 100)
    fold_no += 1

# 5. Performance Metrics
print(f'> Average Accuracy across {n_folds}-folds:
{np.mean(acc_per_fold):.2f}% (+/- {np.std(acc_per_fold):.2f}%)')

# Average Training and Validation Accuracy for 5 folds
plt.plot(np.mean(avg_train_acc, axis=0), label='Average Training
Accuracy')
plt.plot(np.mean(avg_val_acc, axis=0), label='Average Validation
Accuracy')
plt.title('Average Training and Validation Accuracy across 5 folds')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```

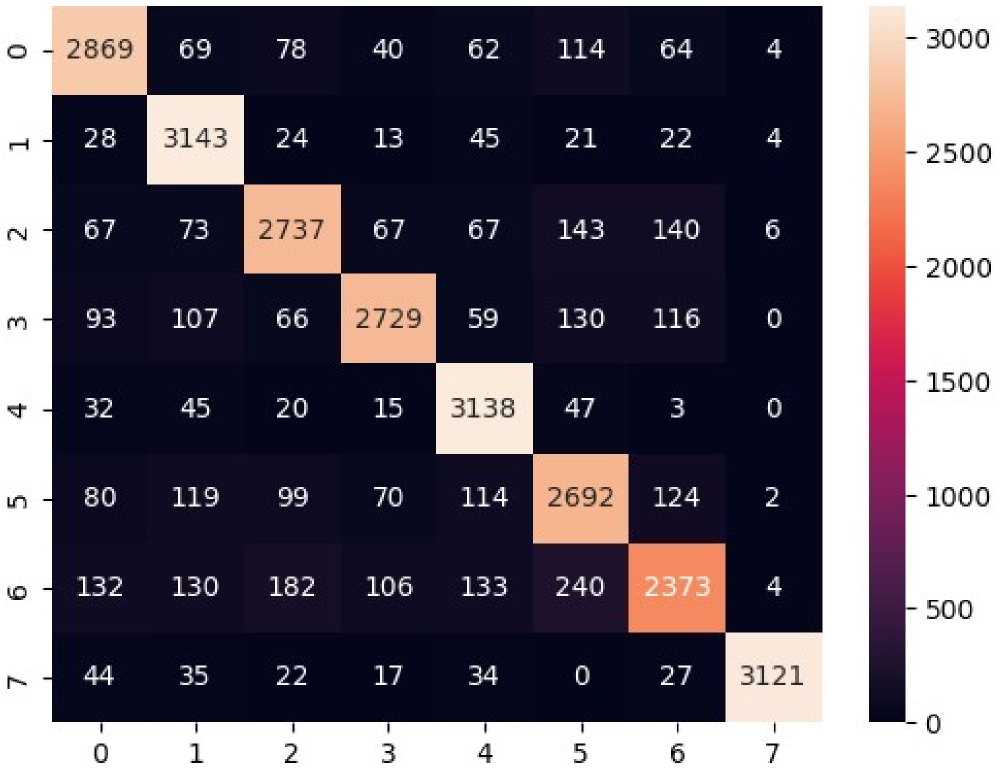
```
# Confusion Matrix
y_pred = model.predict(X_resampled)
y_pred_classes = np.argmax(y_pred, axis=1)
y_true_classes = np.argmax(y_resampled, axis=1)
cm = confusion_matrix(y_true_classes, y_pred_classes)
sns.heatmap(cm, annot=True, fmt='g')
plt.show()

# Classification Report
print(classification_report(y_true_classes, y_pred_classes,
target_names=list(y_dict.keys())))
```

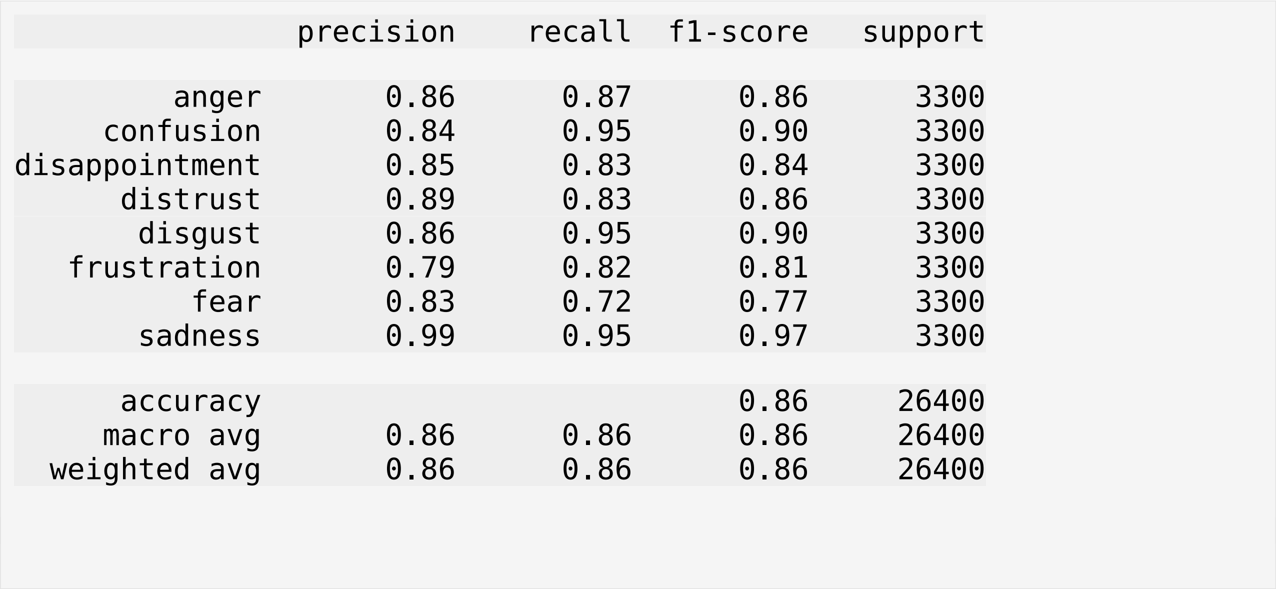
```
/Users/nekdilkhan/miniforge3/lib/python3.12/site-packages/keras/src/
layers/core/embedding.py:90: UserWarning: Argument `input_length` is
deprecated. Just remove it.
  warnings.warn(

> Average Accuracy across 10-folds: 72.54% (+/- 0.85%)
```

```
825/825 ━━━━━━━━━━━━━━━━━━━━ 38s 45ms/step
```



	0	1	2	3	4	5	6	7
0	2869	69	78	40	62	114	64	4
1	28	3143	24	13	45	21	22	4
2	67	73	2737	67	67	143	140	6
3	93	107	66	2729	59	130	116	0
4	32	45	20	15	3138	47	3	0
5	80	119	99	70	114	2692	124	2
6	132	130	182	106	133	240	2373	4
7	44	35	22	17	34	0	27	3121



```
                precision    recall  f1-score   support

         anger       0.86      0.87      0.86      3300
     confusion       0.84      0.95      0.90      3300
disappointment       0.85      0.83      0.84      3300
      distrust       0.89      0.83      0.86      3300
       disgust       0.86      0.95      0.90      3300
   frustration       0.79      0.82      0.81      3300
          fear       0.83      0.72      0.77      3300
       sadness       0.99      0.95      0.97      3300

      accuracy                           0.86     26400
     macro avg       0.86      0.86      0.86     26400
  weighted avg       0.86      0.86      0.86     26400
```