In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
colors = plt.rcParams['axes.prop_cycle'].by_key()['color']
import seaborn as sns
from wordcloud import WordCloud
import re
import pickle

import nltk
from nltk.tokenize import RegexpTokenizer
from nltk.probability import FreqDist
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords

from sklearn.metrics import confusion_matrix
from sklearn.preprocessing import OneHotEncoder, LabelEncoder
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split

import keras
import tensorflow as tf
from keras.layers import GlobalMaxPool1D, Bidirectional, Dropout, Dense, LSTM
from tensorflow.keras import models, layers, Sequential
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint, ReduceLROnPlateau

from keras.utils import to_categorical
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.utils import to_categorical

%run twitter.py
%run plot.py
```

DATA COLLECTION

The dataset comes from CrowdFlower via data.world (Links to an external site.). Human raters rated the sentiment in over 9,000 Tweets as positive, negative, or neithe

https://data.world/crowdflower/brands-and-product-emotions

In [2]:

```
df = pd.read_csv('./data/judge-1377884607_tweet_product_company.csv')
```

In [3]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9093 entries, 0 to 9092
Data columns (total 3 columns):
 #   Column                                              Non-Null Count  Dtype
---  ------                                              --------------  -----
 0   tweet_text                                          9092 non-null   object
 1   emotion_in_tweet_is_directed_at                     3291 non-null   object
 2   is_there_an_emotion_directed_at_a_brand_or_product  9093 non-null   object
dtypes: object(3)
memory usage: 213.2+ KB
```

In [4]:

```
df.head()
```

Out[4]:

	tweet_text	emotion_in_tweet_is_directed_at	is_there_an_emotion_directed_at_a_brand_or_product
0	.@wesley83 I have a 3G iPhone. After 3 hrs twe...	iPhone	Negative emotion
1	@jessedee Know about @fludapp ? Awesome iPad/i...	iPad or iPhone App	Positive emotion
2	@swonderlin Can not wait for #iPad 2 also. The...	iPad	Positive emotion
3	@sxsw I hope this year's festival isn't as cra...	iPad or iPhone App	Negative emotion
4	@sxtxstate great stuff on Fri #SXSW: Marissa M...	Google	Positive emotion

DATA SCRUBBING

In [5]:

```
df.drop(labels=['emotion_in_tweet_is_directed_at'], inplace=True, axis=1)
```

In [6]:

```
df.columns=['tweet_raw', 'sentiment']
```

In [7]:

```
df.sentiment = df.sentiment.apply(lambda x: 'neutral' if x =='No emotion toward brand or product' else x.lower())
df.sentiment = df.sentiment.apply(lambda x: 'positive' if x =='positive emotion' else x.lower())
df.sentiment = df.sentiment.apply(lambda x: 'negative' if x =='negative emotion' else x.lower())
```

In [8]:

```
df.dropna(inplace=True)
```

In [9]:

```
df.head()
```

Out[9]:

	tweet_raw	sentiment
0	.@wesley83 I have a 3G iPhone. After 3 hrs twe...	negative
1	@jessedee Know about @fludapp ? Awesome iPad/i...	positive
2	@swonderlin Can not wait for #iPad 2 also. The...	positive
3	@sxsw I hope this year's festival isn't as cra...	negative
4	@sxtxstate great stuff on Fri #SXSW: Marissa M...	positive

In [10]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9092 entries, 0 to 9092
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   tweet_raw  9092 non-null   object
 1   sentiment  9092 non-null   object
dtypes: object(2)
memory usage: 213.1+ KB
```

PREPROCESSING

In [11]:

```
nltk.download('words')
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
```

```
[nltk_data] Downloading package words to /Users/boula/nltk_data...
[nltk_data]   Package words is already up-to-date!
[nltk_data] Downloading package stopwords to /Users/boula/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /Users/boula/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package wordnet to /Users/boula/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

Out[11]: True

In [11]:

```
stop_words=stopwords.words('english')
```

In [12]:

```
urls = url_extractor(df.tweet_raw)
```

In [13]:

```
hashtag_list = hashtags(df.tweet_raw)
```

In [14]:

```
df['tweet'] = CleanUp(df.tweet_raw)
```

In [15]:

```
tokenizer = RegexpTokenizer('[a-zA-Z0-9]+')
df.tweet  =  df.tweet.apply(lambda x: tokenizer.tokenize(x))
```

In [16]:

```
stop_words_en = set(stopwords.words('english'))
df.tweet      = df.tweet.apply(lambda tweet: [word for word in tweet if word not in stop_words_en])
```

In [17]:

```
lemmatizer  = WordNetLemmatizer()
df.tweet    = df.tweet.apply(lambda tweet: [lemmatizer.lemmatize(word) for word in tweet])
```

In [18]:

```
df.tweet = df.tweet.apply(lambda tweet: ' '.join(tweet))
```

FEATURES ENGINEERING

In [19]:

```
twitt = Twitter(df.tweet)
```

In [20]:

```
df['lexical_diversity']= twitt.Lexical_Diversity
```

In [21]:

```
df['word_count'] = twitt.WordsCount
```

EXPLORATORY DATA ANALYSIS

In [23]:

```
tdf = df.groupby(['sentiment'], as_index=False).count().reset_index(drop=True).\
sort_values(by='tweet_raw', ascending=False).style.background_gradient(cmap='Purples')
tdf
```

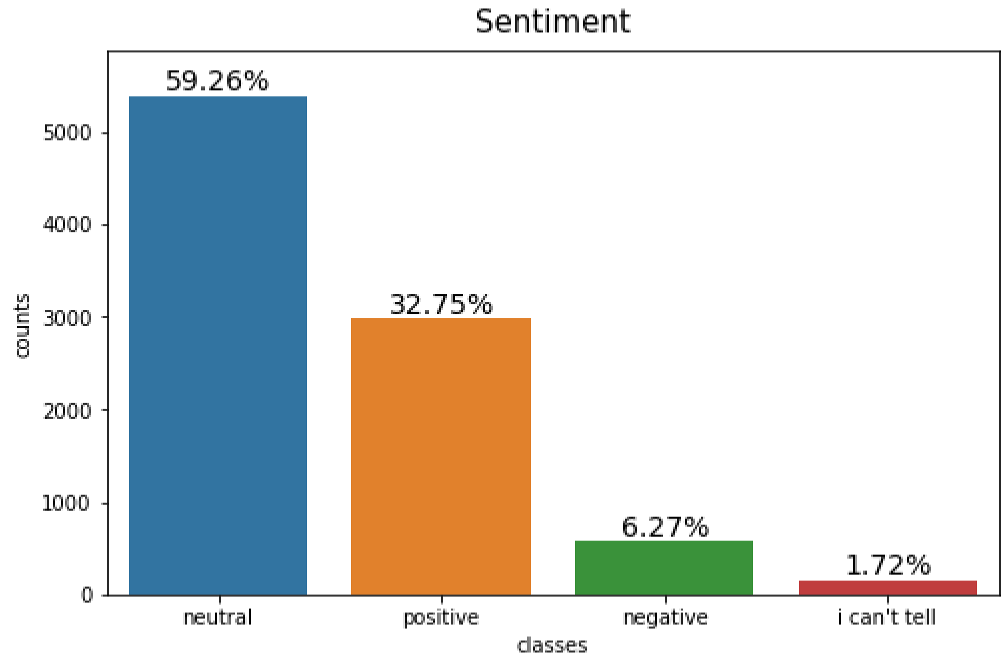
Out[23]:

	sentiment	tweet_raw	tweet	lexical_diversity	word_count
2	neutral	5388	5388	5388	5388

	sentiment	tweet_raw	tweet	lexical_diversity	word_count
3	positive	2978	2978	2978	2978
1	negative	570	570	570	570
0	i can't tell	156	156	156	156

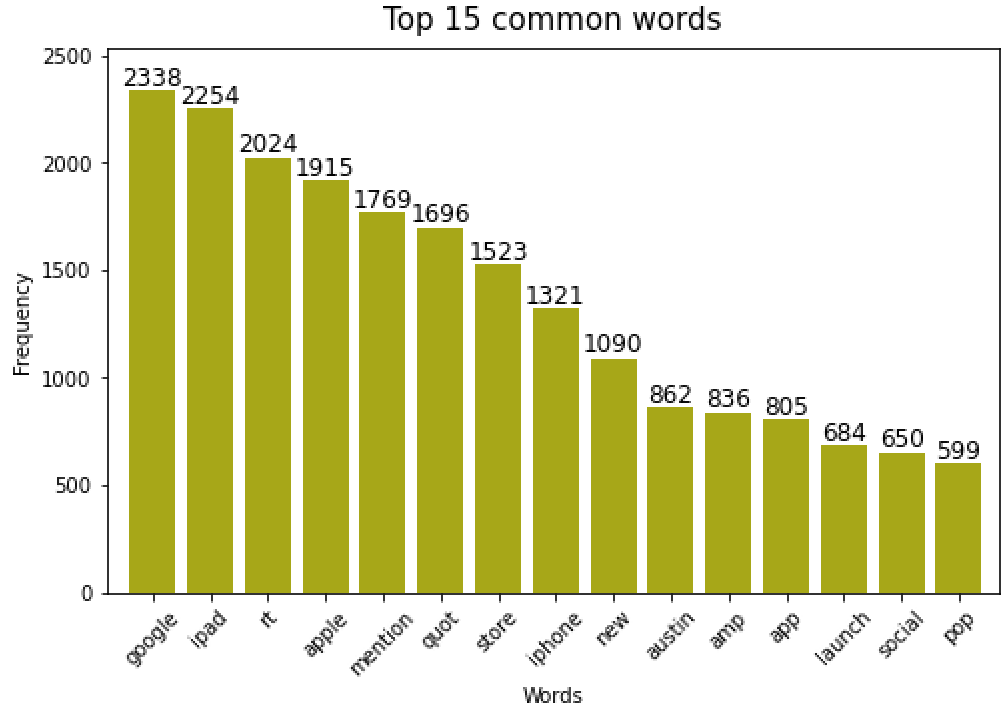
In [24]:

```
fig, ax = plt.subplots(figsize=(8,5))
g = sns.barplot(x=df.sentiment.value_counts().index,
                y=df.sentiment.value_counts().values, ax=ax)
for index, row in df.sentiment.value_counts().iteritems():
    g.text(x=df.sentiment.value_counts().index.get_loc(index),
           y=row,
           s='{0:.2f}%'.format(row/df.shape[0]*100),
           color='black', fontsize=14, horizontalalignment='center', va='bottom')
ax.set_ylim((0, df.sentiment.value_counts().max()+500))
ax.set_title("Sentiment", fontsize=15, ha='center', va='bottom')
ax.set_xlabel('classes')
ax.set_ylabel('counts')
plt.savefig('./img/sentiment_classes.jpg',format="jpg")
plt.show()
```



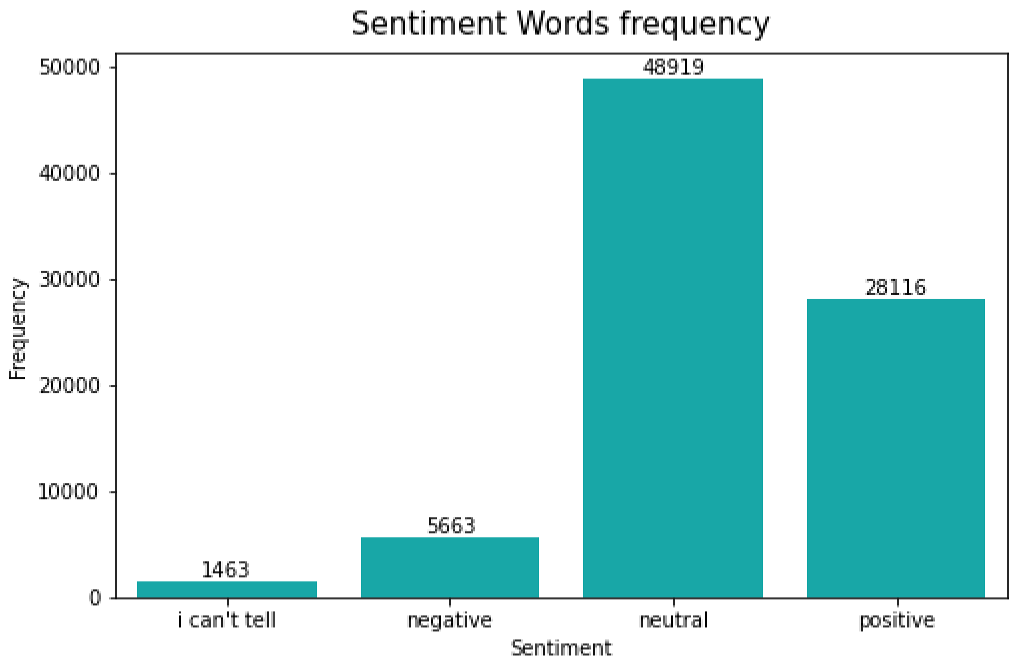
In [25]:

```
df_freq = pd.DataFrame(twitt.FrequencyDist.items(), columns=['word','counts'])
fig, ax = plt.subplots(figsize=(8,5))
sns_word_freq = sns.barplot(x=df_freq.word.head(15), y=df_freq.counts.head(15), data=df_freq, color='y')
for idx, row in df_freq.head(15).iterrows():
    sns_word_freq.text(x=row.name, y=row.counts, s=round(row.counts),
                       color='black', va='bottom',  ha="center", fontsize=12)
ax.set_xlim((-0.8, 14.7))
ax.set_ylim((0, df_freq.head(15).counts.max()+200))
ax.set_title('Top 15 common words', fontsize=15, ha='center', va='bottom')
ax.set_xlabel('Words')
ax.set_ylabel('Frequency')
plt.xticks(rotation=45)
plt.savefig('./img/word_frequency.jpg',format="jpg")
plt.show()
```



In [26]:

```
df_word_count = df.groupby('sentiment', as_index=False).agg({'word_count': 'sum'})
fig, ax = plt.subplots(figsize=(8,5))
sns_word_count = sns.barplot(x='sentiment', y='word_count', data=df_word_count, color='c')
for idx, row in df_word_count.iterrows():
    sns_word_count.text(x=row.name, y=row.word_count, s=round(row.word_count),
                        color='black', va='bottom',  ha="center", fontsize=10)
ax.set_title("Sentiment Words frequency", fontsize=15, ha='center', va='bottom')
ax.set_xlabel('Sentiment')
ax.set_ylabel('Frequency')
plt.savefig('./img/word_count_frequency.jpg',format="jpg")
plt.show()
```



In [27]:

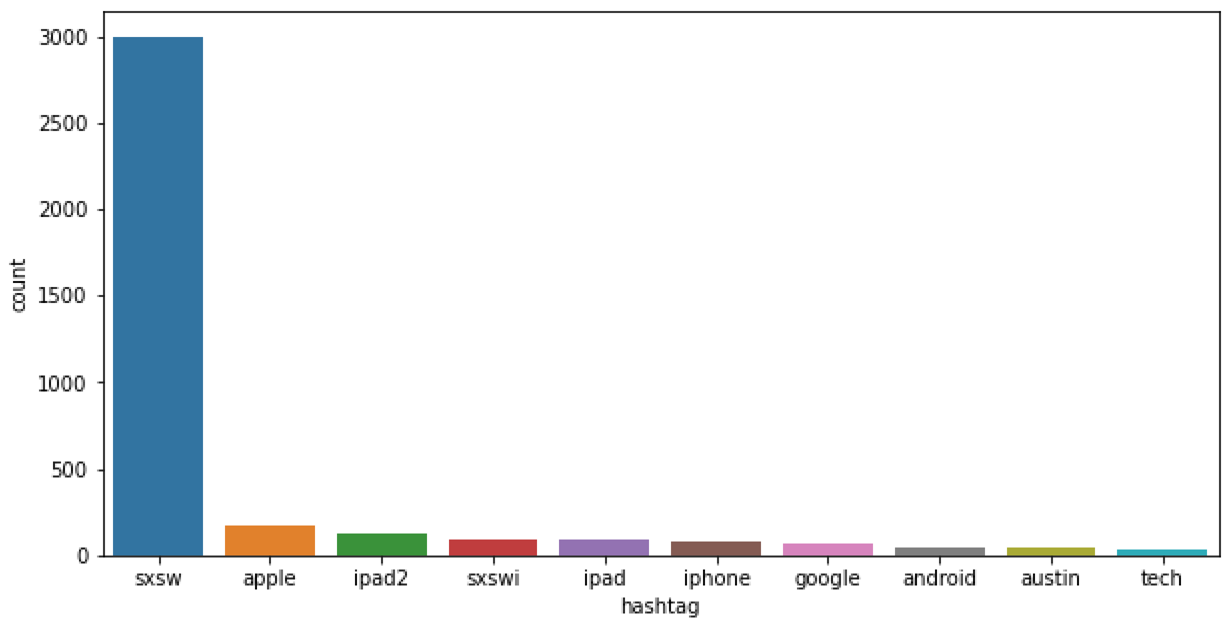
```
df_hashtag_all = pd.DataFrame(hashtags(df.tweet_raw).items(), columns=['hashtag', 'count'])
df_hashtag_pos = pd.DataFrame(hashtags(df[(df.sentiment=='positive')].tweet_raw).items(), columns=['hashtag','count'])
df_hashtag_neg = pd.DataFrame(hashtags(df[(df.sentiment=='negative')].tweet_raw).items(), columns=['hashtag','count'])
```

In [28]:

```
fig, ax = plt.subplots(figsize=(10,5))

# sns.barplot(x='hashtag', y='count', data=df_hashtag_all.head(10))
sns.barplot(x='hashtag', y='count', data=df_hashtag_pos.head(10))
# sns.barplot(x='hashtag', y='count', data=df_hashtag_all.head(10))
```

Out[28]: <AxesSubplot:xlabel='hashtag', ylabel='count'>



In [29]:

```
positive = Twitter(df[(df.sentiment=='positive')].tweet)

WC1= WordCloud(
    max_font_size=150,
    background_color="white",
    ranks_only=True
).generate(text= ' '.join([k for k in positive.FrequencyDist.keys()]) )

plt.figure(figsize=(10,7), facecolor='w')
plt.imshow(WC1, interpolation='bilinear')
plt.axis("off")
plt.tight_layout(pad=0)
plt.savefig('./img/word_cloud_positive.jpg',format="jpg")
plt.show()
```

In [30]:

```
negative = Twitter(df[(df.sentiment=='negative')].tweet)
```

```python
WC2= WordCloud(
    max_font_size=150,
    background_color="white",
    ranks_only=True
).generate(text= ' '.join([k for k in negative.FrequencyDist.keys()]) )

plt.figure(figsize=(10,7), facecolor='w')
plt.imshow(WC2, interpolation='bilinear')
plt.axis("off")
plt.tight_layout(pad=0)
plt.savefig('./img/word_cloud_negative.jpg',format="jpg")
plt.show()
```

Modeling

In [22]:
```python
ohe = OneHotEncoder(sparse=False)
Y_ohe = ohe.fit_transform(df.sentiment.values.reshape(-1,1))
print(f"{'Y_ohe Categories:':20}{ohe.categories_[0]}")
print(f"{'Y_ohe Shape':20}{Y_ohe.shape}")
```

```
Y_ohe Categories:   ["i can't tell" 'negative' 'neutral' 'positive']
Y_ohe Shape         (9092, 4)
```

In [24]:
```python
Test_Size = int(df.shape[0]*.15)

tfidfVectorizer = TfidfVectorizer(max_features=10_000,
                                  norm='l1',
                                  strip_accents='ascii',
                                  stop_words=stop_words,
                                  analyzer='word',
                                  ngram_range=(1,1))
X_train, X_test, y_train, y_test = train_test_split(df.tweet, Y_ohe, test_size=Test_Size, random_state=67)
X_train   = tfidfVectorizer.fit_transform(X_train)
X_train   = X_train.toarray()
X_test    = tfidfVectorizer.transform(X_test)
X_test    = X_test.toarray()
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=Test_Size, random_state=67)

print(f"Train\n\t{'X :':5}{X_train.shape}\n\t{'Y :':5}{y_train.shape}")
print(f"Test\n\t{'X :':5}{X_test.shape}\n\t{'Y :':5}{y_test.shape}")
print(f"Validation\n\t{'X :':5}{X_val.shape}\n\t{'Y :':5}{y_val.shape}")
```

```
Train
        X :  (6366, 7045)
        Y :  (6366, 4)
Test
        X :  (1363, 7045)
        Y :  (1363, 4)
Validation
        X :  (1363, 7045)
        Y :  (1363, 4)
```

In [25]:
```python
eStop_Val_Loss = EarlyStopping(monitor='val_loss', min_delta=0.0001, patience=24,
                               verbose=1,mode='min',baseline=0.999,restore_best_weights=False)
eStop_Loss     = EarlyStopping(monitor='loss', min_delta=0.001, patience=24,
                               verbose=1,mode='min',baseline=0.999,restore_best_weights=False)
eStop_AUC      = EarlyStopping(monitor='auc', min_delta=0.001, patience=24,
                               verbose=1, mode='max', baseline=0.999, restore_best_weights=False)
eStop_Val_AUC  = EarlyStopping(monitor='val_auc', min_delta=0.001, patience=50,
                               verbose=1, mode='max', baseline=0.999, restore_best_weights=False)
```

In [26]:
```python
METRICS1 = ['AUC']
EPOCHS = 128
BATCH_SIZE = 2048
```

Baseline model

In [78]:

```
keras.backend.clear_session()
```

In [94]:

```
model1 = make_model(metrics=METRICS1,
                    input_shape=X_train.shape[1],
                    output_shape=y_train.shape[1],
                    bias_initializer=tf.keras.initializers.zeros,
                    kernel_initializer=keras.initializers.GlorotUniform(),
                    optimizer=keras.optimizers.Adam(learning_rate=0.001)
                   )
model1.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input32 (Dense)              (None, 64)                450944    
_________________________________________________________________
dropout (Dropout)            (None, 64)                0         
_________________________________________________________________
dense16 (Dense)              (None, 16)                1040      
_________________________________________________________________
dense8 (Dense)               (None, 8)                 136       
_________________________________________________________________
output4 (Dense)              (None, 4)                 36        
=================================================================
Total params: 452,156
Trainable params: 452,156
Non-trainable params: 0
_________________________________________________________________
```

In [95]:

```
history = model1.fit(X_train,
                    y_train,
                    validation_data=(X_val, y_val),
                    validation_split=0.1,
                    validation_batch_size=1,
                    epochs=EPOCHS,
                    batch_size=X_train.shape[0],
#                     callbacks=[eStop_val_loss],
                    use_multiprocessing = True,
                    workers=8
                    )
```

```
Epoch 1/128
1/1 [==============================] - 1s 1s/step - loss: 1.3742 - auc: 0.5652 - val_loss: 1.3620 - val_auc: 0.5780
Epoch 2/128
1/1 [==============================] - 0s 454ms/step - loss: 1.3707 - auc: 0.5664 - val_loss: 1.3589 - val_auc: 0.5784
Epoch 3/128
1/1 [==============================] - 0s 439ms/step - loss: 1.3675 - auc: 0.5667 - val_loss: 1.3554 - val_auc: 0.5795
Epoch 4/128
1/1 [==============================] - 0s 431ms/step - loss: 1.3642 - auc: 0.5697 - val_loss: 1.3517 - val_auc: 0.5787
Epoch 5/128
1/1 [==============================] - 0s 421ms/step - loss: 1.3607 - auc: 0.5701 - val_loss: 1.3477 - val_auc: 0.5790
Epoch 6/128
1/1 [==============================] - 0s 417ms/step - loss: 1.3569 - auc: 0.5699 - val_loss: 1.3436 - val_auc: 0.5793
Epoch 7/128
1/1 [==============================] - 0s 447ms/step - loss: 1.3529 - auc: 0.5707 - val_loss: 1.3393 - val_auc: 0.5815
Epoch 8/128
1/1 [==============================] - 0s 421ms/step - loss: 1.3488 - auc: 0.5730 - val_loss: 1.3349 - val_auc: 0.5806
Epoch 9/128
1/1 [==============================] - 0s 423ms/step - loss: 1.3447 - auc: 0.5723 - val_loss: 1.3303 - val_auc: 0.5800
Epoch 10/128
1/1 [==============================] - 0s 424ms/step - loss: 1.3403 - auc: 0.5729 - val_loss: 1.3256 - val_auc: 0.5847
Epoch 11/128
1/1 [==============================] - 0s 423ms/step - loss: 1.3356 - auc: 0.5738 - val_loss: 1.3207 - val_auc: 0.5833
Epoch 12/128
1/1 [==============================] - 0s 430ms/step - loss: 1.3310 - auc: 0.5741 - val_loss: 1.3157 - val_auc: 0.5852
Epoch 13/128
1/1 [==============================] - 0s 416ms/step - loss: 1.3263 - auc: 0.5754 - val_loss: 1.3105 - val_auc: 0.5853
Epoch 14/128
1/1 [==============================] - 0s 429ms/step - loss: 1.3214 - auc: 0.5749 - val_loss: 1.3053 - val_auc: 0.5836
Epoch 15/128
1/1 [==============================] - 0s 427ms/step - loss: 1.3162 - auc: 0.5756 - val_loss: 1.2999 - val_auc: 0.5867
Epoch 16/128
1/1 [==============================] - 0s 426ms/step - loss: 1.3111 - auc: 0.5761 - val_loss: 1.2944 - val_auc: 0.5863
Epoch 17/128
1/1 [==============================] - 0s 426ms/step - loss: 1.3054 - auc: 0.5773 - val_loss: 1.2889 - val_auc: 0.5879
Epoch 18/128
1/1 [==============================] - 0s 419ms/step - loss: 1.3001 - auc: 0.5782 - val_loss: 1.2831 - val_auc: 0.5886
Epoch 19/128
1/1 [==============================] - 0s 420ms/step - loss: 1.2944 - auc: 0.5875 - val_loss: 1.2773 - val_auc: 0.5925
Epoch 20/128
1/1 [==============================] - 0s 418ms/step - loss: 1.2889 - auc: 0.6124 - val_loss: 1.2713 - val_auc: 0.6417
Epoch 21/128
1/1 [==============================] - 0s 420ms/step - loss: 1.2827 - auc: 0.6572 - val_loss: 1.2652 - val_auc: 0.7353
Epoch 22/128
1/1 [==============================] - 0s 435ms/step - loss: 1.2767 - auc: 0.6989 - val_loss: 1.2589 - val_auc: 0.7733
Epoch 23/128
1/1 [==============================] - 0s 424ms/step - loss: 1.2704 - auc: 0.7312 - val_loss: 1.2525 - val_auc: 0.7809
Epoch 24/128
1/1 [==============================] - 0s 419ms/step - loss: 1.2643 - auc: 0.7451 - val_loss: 1.2460 - val_auc: 0.7811
Epoch 25/128
1/1 [==============================] - 0s 422ms/step - loss: 1.2575 - auc: 0.7519 - val_loss: 1.2393 - val_auc: 0.7809
Epoch 26/128
1/1 [==============================] - 0s 417ms/step - loss: 1.2511 - auc: 0.7539 - val_loss: 1.2325 - val_auc: 0.7796
Epoch 27/128
1/1 [==============================] - 0s 417ms/step - loss: 1.2441 - auc: 0.7548 - val_loss: 1.2255 - val_auc: 0.7801
Epoch 28/128
1/1 [==============================] - 0s 411ms/step - loss: 1.2376 - auc: 0.7537 - val_loss: 1.2184 - val_auc: 0.7828
Epoch 29/128
1/1 [==============================] - 0s 423ms/step - loss: 1.2301 - auc: 0.7552 - val_loss: 1.2111 - val_auc: 0.7825
Epoch 30/128
1/1 [==============================] - 0s 425ms/step - loss: 1.2229 - auc: 0.7549 - val_loss: 1.2037 - val_auc: 0.7806
```

```
Epoch 31/128
1/1 [==============================] - 0s 424ms/step - loss: 1.2169 - auc: 0.7568 - val_loss: 1.1962 - val_auc: 0.7809
Epoch 32/128
1/1 [==============================] - 0s 428ms/step - loss: 1.2086 - auc: 0.7648 - val_loss: 1.1885 - val_auc: 0.7827
Epoch 33/128
1/1 [==============================] - 0s 426ms/step - loss: 1.2016 - auc: 0.7738 - val_loss: 1.1808 - val_auc: 0.7953
Epoch 34/128
1/1 [==============================] - 0s 424ms/step - loss: 1.1938 - auc: 0.7878 - val_loss: 1.1729 - val_auc: 0.8161
Epoch 35/128
1/1 [==============================] - 0s 425ms/step - loss: 1.1871 - auc: 0.8011 - val_loss: 1.1649 - val_auc: 0.8418
Epoch 36/128
1/1 [==============================] - 0s 427ms/step - loss: 1.1782 - auc: 0.8175 - val_loss: 1.1568 - val_auc: 0.8603
Epoch 37/128
1/1 [==============================] - 0s 426ms/step - loss: 1.1705 - auc: 0.8299 - val_loss: 1.1486 - val_auc: 0.8633
Epoch 38/128
1/1 [==============================] - 0s 423ms/step - loss: 1.1633 - auc: 0.8360 - val_loss: 1.1404 - val_auc: 0.8648
Epoch 39/128
1/1 [==============================] - 0s 426ms/step - loss: 1.1557 - auc: 0.8417 - val_loss: 1.1322 - val_auc: 0.8631
Epoch 40/128
1/1 [==============================] - 0s 422ms/step - loss: 1.1470 - auc: 0.8464 - val_loss: 1.1239 - val_auc: 0.8643
Epoch 41/128
1/1 [==============================] - 0s 424ms/step - loss: 1.1399 - auc: 0.8459 - val_loss: 1.1157 - val_auc: 0.8633
Epoch 42/128
1/1 [==============================] - 0s 423ms/step - loss: 1.1319 - auc: 0.8458 - val_loss: 1.1074 - val_auc: 0.8635
Epoch 43/128
1/1 [==============================] - 0s 420ms/step - loss: 1.1231 - auc: 0.8472 - val_loss: 1.0992 - val_auc: 0.8624
Epoch 44/128
1/1 [==============================] - 0s 432ms/step - loss: 1.1164 - auc: 0.8465 - val_loss: 1.0909 - val_auc: 0.8654
Epoch 45/128
1/1 [==============================] - 0s 428ms/step - loss: 1.1068 - auc: 0.8496 - val_loss: 1.0828 - val_auc: 0.8639
Epoch 46/128
1/1 [==============================] - 0s 424ms/step - loss: 1.0999 - auc: 0.8482 - val_loss: 1.0746 - val_auc: 0.8645
Epoch 47/128
1/1 [==============================] - 0s 428ms/step - loss: 1.0923 - auc: 0.8487 - val_loss: 1.0665 - val_auc: 0.8627
Epoch 48/128
1/1 [==============================] - 0s 424ms/step - loss: 1.0848 - auc: 0.8485 - val_loss: 1.0585 - val_auc: 0.8649
Epoch 49/128
1/1 [==============================] - 0s 429ms/step - loss: 1.0779 - auc: 0.8492 - val_loss: 1.0506 - val_auc: 0.8661
Epoch 50/128
1/1 [==============================] - 0s 429ms/step - loss: 1.0696 - auc: 0.8482 - val_loss: 1.0428 - val_auc: 0.8656
Epoch 51/128
1/1 [==============================] - 0s 420ms/step - loss: 1.0629 - auc: 0.8486 - val_loss: 1.0350 - val_auc: 0.8628
Epoch 52/128
1/1 [==============================] - 0s 419ms/step - loss: 1.0545 - auc: 0.8507 - val_loss: 1.0274 - val_auc: 0.8633
Epoch 53/128
1/1 [==============================] - 0s 423ms/step - loss: 1.0461 - auc: 0.8509 - val_loss: 1.0199 - val_auc: 0.8673
Epoch 54/128
1/1 [==============================] - 0s 425ms/step - loss: 1.0392 - auc: 0.8509 - val_loss: 1.0126 - val_auc: 0.8664
Epoch 55/128
1/1 [==============================] - 0s 425ms/step - loss: 1.0335 - auc: 0.8506 - val_loss: 1.0054 - val_auc: 0.8632
Epoch 56/128
1/1 [==============================] - 0s 430ms/step - loss: 1.0262 - auc: 0.8508 - val_loss: 0.9983 - val_auc: 0.8624
Epoch 57/128
1/1 [==============================] - 0s 428ms/step - loss: 1.0180 - auc: 0.8518 - val_loss: 0.9913 - val_auc: 0.8653
Epoch 58/128
1/1 [==============================] - 0s 422ms/step - loss: 1.0111 - auc: 0.8517 - val_loss: 0.9846 - val_auc: 0.8672
Epoch 59/128
1/1 [==============================] - 0s 419ms/step - loss: 1.0054 - auc: 0.8512 - val_loss: 0.9780 - val_auc: 0.8605
Epoch 60/128
1/1 [==============================] - 0s 418ms/step - loss: 0.9987 - auc: 0.8518 - val_loss: 0.9716 - val_auc: 0.8598
Epoch 61/128
1/1 [==============================] - 0s 426ms/step - loss: 0.9928 - auc: 0.8514 - val_loss: 0.9654 - val_auc: 0.8629
Epoch 62/128
1/1 [==============================] - 0s 424ms/step - loss: 0.9852 - auc: 0.8534 - val_loss: 0.9593 - val_auc: 0.8680
Epoch 63/128
1/1 [==============================] - 0s 425ms/step - loss: 0.9807 - auc: 0.8525 - val_loss: 0.9534 - val_auc: 0.8634
Epoch 64/128
1/1 [==============================] - 0s 426ms/step - loss: 0.9761 - auc: 0.8518 - val_loss: 0.9478 - val_auc: 0.8600
Epoch 65/128
1/1 [==============================] - 0s 429ms/step - loss: 0.9701 - auc: 0.8519 - val_loss: 0.9423 - val_auc: 0.8625
Epoch 66/128
1/1 [==============================] - 0s 423ms/step - loss: 0.9655 - auc: 0.8521 - val_loss: 0.9370 - val_auc: 0.8665
Epoch 67/128
1/1 [==============================] - 0s 416ms/step - loss: 0.9594 - auc: 0.8533 - val_loss: 0.9320 - val_auc: 0.8650
Epoch 68/128
1/1 [==============================] - 0s 430ms/step - loss: 0.9538 - auc: 0.8542 - val_loss: 0.9272 - val_auc: 0.8632
Epoch 69/128
1/1 [==============================] - 0s 435ms/step - loss: 0.9496 - auc: 0.8550 - val_loss: 0.9226 - val_auc: 0.8634
Epoch 70/128
1/1 [==============================] - 0s 434ms/step - loss: 0.9453 - auc: 0.8555 - val_loss: 0.9183 - val_auc: 0.8658
Epoch 71/128
1/1 [==============================] - 0s 427ms/step - loss: 0.9391 - auc: 0.8569 - val_loss: 0.9142 - val_auc: 0.8641
Epoch 72/128
1/1 [==============================] - 0s 421ms/step - loss: 0.9345 - auc: 0.8573 - val_loss: 0.9102 - val_auc: 0.8654
Epoch 73/128
1/1 [==============================] - 0s 435ms/step - loss: 0.9310 - auc: 0.8567 - val_loss: 0.9065 - val_auc: 0.8666
Epoch 74/128
1/1 [==============================] - 0s 422ms/step - loss: 0.9292 - auc: 0.8573 - val_loss: 0.9029 - val_auc: 0.8667
Epoch 75/128
1/1 [==============================] - 0s 431ms/step - loss: 0.9230 - auc: 0.8596 - val_loss: 0.8995 - val_auc: 0.8650
Epoch 76/128
1/1 [==============================] - 0s 423ms/step - loss: 0.9194 - auc: 0.8592 - val_loss: 0.8963 - val_auc: 0.8663
Epoch 77/128
1/1 [==============================] - 0s 427ms/step - loss: 0.9172 - auc: 0.8589 - val_loss: 0.8932 - val_auc: 0.8675
Epoch 78/128
1/1 [==============================] - 0s 427ms/step - loss: 0.9123 - auc: 0.8601 - val_loss: 0.8903 - val_auc: 0.8687
Epoch 79/128
1/1 [==============================] - 0s 434ms/step - loss: 0.9089 - auc: 0.8611 - val_loss: 0.8875 - val_auc: 0.8677
Epoch 80/128
1/1 [==============================] - 0s 425ms/step - loss: 0.9050 - auc: 0.8627 - val_loss: 0.8849 - val_auc: 0.8693
Epoch 81/128
1/1 [==============================] - 0s 425ms/step - loss: 0.9014 - auc: 0.8640 - val_loss: 0.8824 - val_auc: 0.8698
Epoch 82/128
1/1 [==============================] - 0s 428ms/step - loss: 0.8994 - auc: 0.8633 - val_loss: 0.8801 - val_auc: 0.8711
Epoch 83/128
1/1 [==============================] - 0s 422ms/step - loss: 0.8933 - auc: 0.8661 - val_loss: 0.8778 - val_auc: 0.8699
```

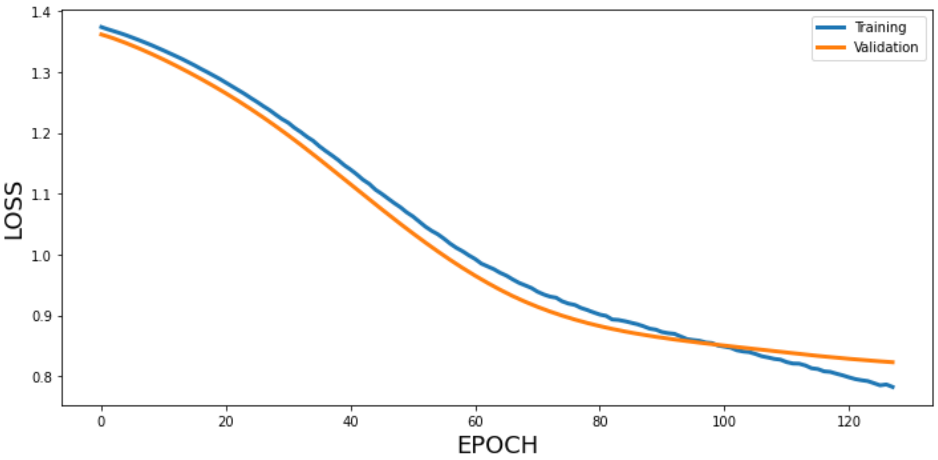
```
Epoch 84/128
1/1 [==============================] - 0s 427ms/step - loss: 0.8926 - auc: 0.8660 - val_loss: 0.8757 - val_auc: 0.8713
Epoch 85/128
1/1 [==============================] - 0s 426ms/step - loss: 0.8906 - auc: 0.8661 - val_loss: 0.8737 - val_auc: 0.8719
Epoch 86/128
1/1 [==============================] - 0s 451ms/step - loss: 0.8881 - auc: 0.8664 - val_loss: 0.8718 - val_auc: 0.8725
Epoch 87/128
1/1 [==============================] - 0s 422ms/step - loss: 0.8856 - auc: 0.8677 - val_loss: 0.8699 - val_auc: 0.8736
Epoch 88/128
1/1 [==============================] - 0s 425ms/step - loss: 0.8823 - auc: 0.8682 - val_loss: 0.8682 - val_auc: 0.8736
Epoch 89/128
1/1 [==============================] - 0s 466ms/step - loss: 0.8784 - auc: 0.8703 - val_loss: 0.8665 - val_auc: 0.8747
Epoch 90/128
1/1 [==============================] - 0s 464ms/step - loss: 0.8766 - auc: 0.8703 - val_loss: 0.8649 - val_auc: 0.8744
Epoch 91/128
1/1 [==============================] - 0s 478ms/step - loss: 0.8727 - auc: 0.8710 - val_loss: 0.8633 - val_auc: 0.8749
Epoch 92/128
1/1 [==============================] - 0s 446ms/step - loss: 0.8708 - auc: 0.8731 - val_loss: 0.8619 - val_auc: 0.8763
Epoch 93/128
1/1 [==============================] - 0s 457ms/step - loss: 0.8697 - auc: 0.8728 - val_loss: 0.8604 - val_auc: 0.8763
Epoch 94/128
1/1 [==============================] - 1s 520ms/step - loss: 0.8654 - auc: 0.8730 - val_loss: 0.8591 - val_auc: 0.8770
Epoch 95/128
1/1 [==============================] - 0s 447ms/step - loss: 0.8611 - auc: 0.8758 - val_loss: 0.8577 - val_auc: 0.8771
Epoch 96/128
1/1 [==============================] - 0s 449ms/step - loss: 0.8595 - auc: 0.8769 - val_loss: 0.8564 - val_auc: 0.8779
Epoch 97/128
1/1 [==============================] - 0s 434ms/step - loss: 0.8584 - auc: 0.8752 - val_loss: 0.8552 - val_auc: 0.8779
Epoch 98/128
1/1 [==============================] - 0s 427ms/step - loss: 0.8555 - auc: 0.8763 - val_loss: 0.8540 - val_auc: 0.8778
Epoch 99/128
1/1 [==============================] - 0s 429ms/step - loss: 0.8546 - auc: 0.8773 - val_loss: 0.8527 - val_auc: 0.8785
Epoch 100/128
1/1 [==============================] - 0s 417ms/step - loss: 0.8504 - auc: 0.8781 - val_loss: 0.8516 - val_auc: 0.8787
Epoch 101/128
1/1 [==============================] - 0s 419ms/step - loss: 0.8487 - auc: 0.8787 - val_loss: 0.8504 - val_auc: 0.8787
Epoch 102/128
1/1 [==============================] - 0s 421ms/step - loss: 0.8469 - auc: 0.8797 - val_loss: 0.8492 - val_auc: 0.8796
Epoch 103/128
1/1 [==============================] - 0s 422ms/step - loss: 0.8427 - auc: 0.8813 - val_loss: 0.8481 - val_auc: 0.8793
Epoch 104/128
1/1 [==============================] - 0s 428ms/step - loss: 0.8408 - auc: 0.8810 - val_loss: 0.8470 - val_auc: 0.8795
Epoch 105/128
1/1 [==============================] - 0s 429ms/step - loss: 0.8397 - auc: 0.8810 - val_loss: 0.8459 - val_auc: 0.8796
Epoch 106/128
1/1 [==============================] - 0s 417ms/step - loss: 0.8366 - auc: 0.8819 - val_loss: 0.8447 - val_auc: 0.8795
Epoch 107/128
1/1 [==============================] - 0s 427ms/step - loss: 0.8330 - auc: 0.8826 - val_loss: 0.8436 - val_auc: 0.8797
Epoch 108/128
1/1 [==============================] - 0s 418ms/step - loss: 0.8309 - auc: 0.8835 - val_loss: 0.8425 - val_auc: 0.8800
Epoch 109/128
1/1 [==============================] - 0s 416ms/step - loss: 0.8286 - auc: 0.8839 - val_loss: 0.8414 - val_auc: 0.8798
Epoch 110/128
1/1 [==============================] - 0s 430ms/step - loss: 0.8273 - auc: 0.8844 - val_loss: 0.8402 - val_auc: 0.8798
Epoch 111/128
1/1 [==============================] - 0s 427ms/step - loss: 0.8233 - auc: 0.8861 - val_loss: 0.8391 - val_auc: 0.8804
Epoch 112/128
1/1 [==============================] - 0s 422ms/step - loss: 0.8209 - auc: 0.8868 - val_loss: 0.8380 - val_auc: 0.8804
Epoch 113/128
1/1 [==============================] - 0s 422ms/step - loss: 0.8206 - auc: 0.8863 - val_loss: 0.8368 - val_auc: 0.8804
Epoch 114/128
1/1 [==============================] - 0s 412ms/step - loss: 0.8178 - auc: 0.8859 - val_loss: 0.8356 - val_auc: 0.8804
Epoch 115/128
1/1 [==============================] - 0s 413ms/step - loss: 0.8132 - auc: 0.8882 - val_loss: 0.8345 - val_auc: 0.8804
Epoch 116/128
1/1 [==============================] - 0s 413ms/step - loss: 0.8119 - auc: 0.8874 - val_loss: 0.8335 - val_auc: 0.8805
Epoch 117/128
1/1 [==============================] - 0s 404ms/step - loss: 0.8081 - auc: 0.8886 - val_loss: 0.8325 - val_auc: 0.8807
Epoch 118/128
1/1 [==============================] - 0s 405ms/step - loss: 0.8072 - auc: 0.8896 - val_loss: 0.8315 - val_auc: 0.8804
Epoch 119/128
1/1 [==============================] - 0s 423ms/step - loss: 0.8042 - auc: 0.8900 - val_loss: 0.8305 - val_auc: 0.8810
Epoch 120/128
1/1 [==============================] - 0s 429ms/step - loss: 0.8015 - auc: 0.8914 - val_loss: 0.8296 - val_auc: 0.8809
Epoch 121/128
1/1 [==============================] - 0s 394ms/step - loss: 0.7983 - auc: 0.8922 - val_loss: 0.8287 - val_auc: 0.8812
Epoch 122/128
1/1 [==============================] - 0s 389ms/step - loss: 0.7954 - auc: 0.8924 - val_loss: 0.8278 - val_auc: 0.8811
Epoch 123/128
1/1 [==============================] - 0s 386ms/step - loss: 0.7935 - auc: 0.8932 - val_loss: 0.8270 - val_auc: 0.8808
Epoch 124/128
1/1 [==============================] - 0s 386ms/step - loss: 0.7921 - auc: 0.8938 - val_loss: 0.8261 - val_auc: 0.8809
Epoch 125/128
1/1 [==============================] - 0s 399ms/step - loss: 0.7887 - auc: 0.8945 - val_loss: 0.8253 - val_auc: 0.8810
Epoch 126/128
1/1 [==============================] - 0s 393ms/step - loss: 0.7851 - auc: 0.8962 - val_loss: 0.8245 - val_auc: 0.8812
Epoch 127/128
1/1 [==============================] - 0s 394ms/step - loss: 0.7865 - auc: 0.8945 - val_loss: 0.8238 - val_auc: 0.8807
Epoch 128/128
1/1 [==============================] - 0s 397ms/step - loss: 0.7824 - auc: 0.8960 - val_loss: 0.8231 - val_auc: 0.8812
```

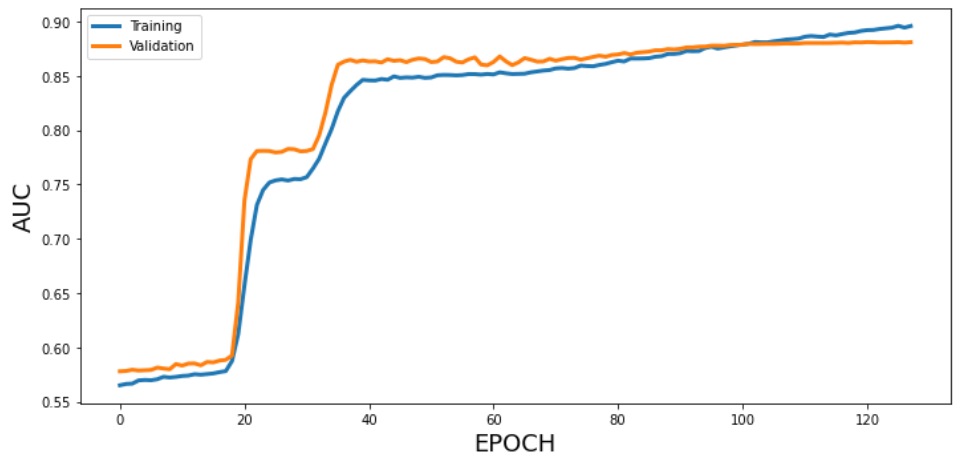
In [101…

```
model1.save_weights(filepath='./tfidf_model/model1')
```

In [100…

```
plot_history(history)
```





Train a model with class weights

Now try re-training and evaluating the model with class weights to see how that affects the predictions.

In [ ]:

```
%run twitter.py
```

In [104…

```
model1 = make_model(metrics=METRICS1,
                    input_shape=X_train.shape[1],
                    output_shape=y_train.shape[1],
                    bias_initializer=tf.keras.initializers.Zeros(),
                    kernel_initializer=keras.initializers.GlorotUniform(),
                    optimizer=keras.optimizers.Adam(learning_rate=0.001),
                    output_bias=baias
                   )
model1.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input32 (Dense)              (None, 64)                450944
_________________________________________________________________
dropout_2 (Dropout)          (None, 64)                0
_________________________________________________________________
dense16 (Dense)              (None, 16)                1040
_________________________________________________________________
dense8 (Dense)               (None, 8)                 136
_________________________________________________________________
output4 (Dense)              (None, 4)                 36
=================================================================
Total params: 452,156
Trainable params: 452,156
Non-trainable params: 0
_________________________________________________________________
```

In [105…

```
history = model1.fit(X_train,
                    y_train,
                    validation_data=(X_val, y_val),
                    validation_split=0.1,
                    validation_batch_size=1,
                    epochs=EPOCHS,
                    batch_size=BATCH_SIZE,
                    #callbacks=[eStop_LOSS],
                    use_multiprocessing = True,
                    workers=8
                    )
```

```
Epoch 1/128
3/3 [==============================] - 1s 240ms/step - loss: 14.1519 - auc: 0.3446 - val_loss: 14.2370 - val_auc: 0.3417
Epoch 2/128
3/3 [==============================] - 0s 162ms/step - loss: 14.1389 - auc: 0.3448 - val_loss: 14.2242 - val_auc: 0.3417
Epoch 3/128
3/3 [==============================] - 0s 160ms/step - loss: 14.1296 - auc: 0.3448 - val_loss: 14.2103 - val_auc: 0.3417
Epoch 4/128
3/3 [==============================] - 0s 158ms/step - loss: 14.0901 - auc: 0.3451 - val_loss: 14.1949 - val_auc: 0.3417
Epoch 5/128
3/3 [==============================] - 0s 160ms/step - loss: 14.1158 - auc: 0.3435 - val_loss: 14.1778 - val_auc: 0.3417
Epoch 6/128
3/3 [==============================] - 0s 162ms/step - loss: 14.0654 - auc: 0.3450 - val_loss: 14.1584 - val_auc: 0.3417
Epoch 7/128
3/3 [==============================] - 0s 160ms/step - loss: 14.0435 - auc: 0.3451 - val_loss: 14.1367 - val_auc: 0.3417
Epoch 8/128
3/3 [==============================] - 0s 159ms/step - loss: 14.0149 - auc: 0.3454 - val_loss: 14.1123 - val_auc: 0.3417
Epoch 9/128
3/3 [==============================] - 1s 335ms/step - loss: 14.0109 - auc: 0.3446 - val_loss: 14.0848 - val_auc: 0.3417
Epoch 10/128
3/3 [==============================] - 0s 160ms/step - loss: 13.9818 - auc: 0.3448 - val_loss: 14.0539 - val_auc: 0.3417
Epoch 11/128
3/3 [==============================] - 0s 167ms/step - loss: 13.9431 - auc: 0.3448 - val_loss: 14.0186 - val_auc: 0.3417
Epoch 12/128
3/3 [==============================] - 0s 163ms/step - loss: 13.9121 - auc: 0.3448 - val_loss: 13.9775 - val_auc: 0.3417
Epoch 13/128
3/3 [==============================] - 0s 162ms/step - loss: 13.8612 - auc: 0.3446 - val_loss: 13.9299 - val_auc: 0.3417
Epoch 14/128
3/3 [==============================] - 0s 165ms/step - loss: 13.7943 - auc: 0.3454 - val_loss: 13.8750 - val_auc: 0.3417
Epoch 15/128
3/3 [==============================] - 0s 162ms/step - loss: 13.7851 - auc: 0.3443 - val_loss: 13.8119 - val_auc: 0.3417
Epoch 16/128
3/3 [==============================] - 0s 164ms/step - loss: 13.7063 - auc: 0.3447 - val_loss: 13.7394 - val_auc: 0.3417
Epoch 17/128
3/3 [==============================] - 0s 168ms/step - loss: 13.6386 - auc: 0.3441 - val_loss: 13.6561 - val_auc: 0.3417
Epoch 18/128
```

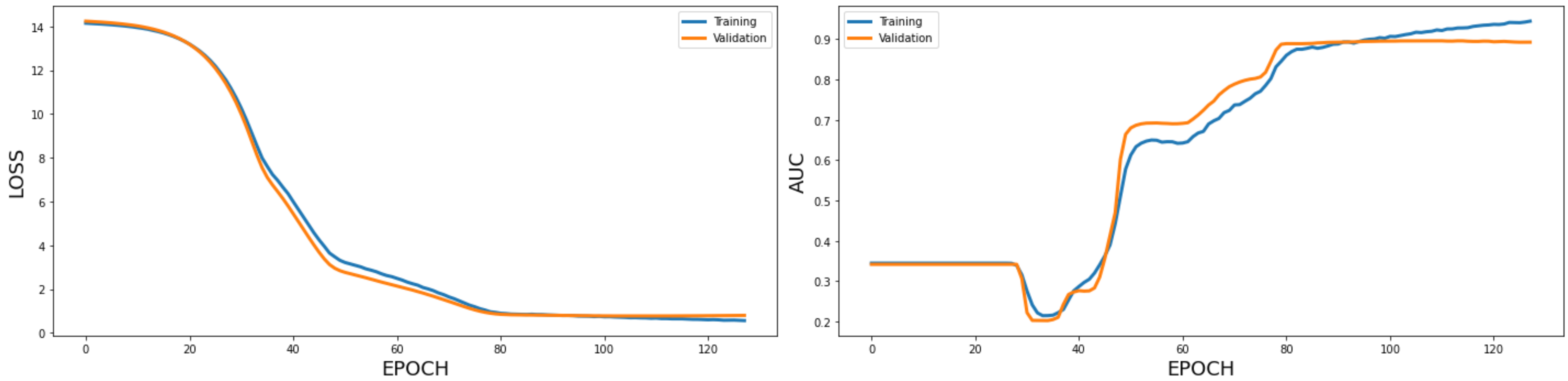
```
3/3 [==============================] - 0s 164ms/step - loss: 13.5463 - auc: 0.3441 - val_loss: 13.5604 - val_auc: 0.3417
Epoch 19/128
3/3 [==============================] - 0s 164ms/step - loss: 13.4533 - auc: 0.3446 - val_loss: 13.4506 - val_auc: 0.3417
Epoch 20/128
3/3 [==============================] - 0s 167ms/step - loss: 13.3419 - auc: 0.3444 - val_loss: 13.3248 - val_auc: 0.3417
Epoch 21/128
3/3 [==============================] - 0s 169ms/step - loss: 13.1940 - auc: 0.3452 - val_loss: 13.1809 - val_auc: 0.3417
Epoch 22/128
3/3 [==============================] - 0s 166ms/step - loss: 13.0540 - auc: 0.3442 - val_loss: 13.0160 - val_auc: 0.3417
Epoch 23/128
3/3 [==============================] - 0s 163ms/step - loss: 12.8691 - auc: 0.3452 - val_loss: 12.8262 - val_auc: 0.3417
Epoch 24/128
3/3 [==============================] - 0s 167ms/step - loss: 12.6980 - auc: 0.3449 - val_loss: 12.6093 - val_auc: 0.3417
Epoch 25/128
3/3 [==============================] - 0s 169ms/step - loss: 12.4975 - auc: 0.3433 - val_loss: 12.3615 - val_auc: 0.3417
Epoch 26/128
3/3 [==============================] - 0s 160ms/step - loss: 12.2161 - auc: 0.3446 - val_loss: 12.0791 - val_auc: 0.3417
Epoch 27/128
3/3 [==============================] - 0s 168ms/step - loss: 11.8950 - auc: 0.3455 - val_loss: 11.7575 - val_auc: 0.3417
Epoch 28/128
3/3 [==============================] - 0s 168ms/step - loss: 11.6023 - auc: 0.3439 - val_loss: 11.3923 - val_auc: 0.3417
Epoch 29/128
3/3 [==============================] - 0s 162ms/step - loss: 11.2292 - auc: 0.3401 - val_loss: 10.9791 - val_auc: 0.3417
Epoch 30/128
3/3 [==============================] - 0s 165ms/step - loss: 10.8021 - auc: 0.3195 - val_loss: 10.5128 - val_auc: 0.3061
Epoch 31/128
3/3 [==============================] - 0s 178ms/step - loss: 10.3195 - auc: 0.2801 - val_loss: 9.9859 - val_auc: 0.2217
Epoch 32/128
3/3 [==============================] - 0s 167ms/step - loss: 9.7755 - auc: 0.2436 - val_loss: 9.3987 - val_auc: 0.2027
Epoch 33/128
3/3 [==============================] - 0s 169ms/step - loss: 9.1940 - auc: 0.2218 - val_loss: 8.7640 - val_auc: 0.2026
Epoch 34/128
3/3 [==============================] - 0s 163ms/step - loss: 8.6311 - auc: 0.2129 - val_loss: 8.1220 - val_auc: 0.2022
Epoch 35/128
3/3 [==============================] - 0s 164ms/step - loss: 8.0558 - auc: 0.2141 - val_loss: 7.5468 - val_auc: 0.2022
Epoch 36/128
3/3 [==============================] - 0s 169ms/step - loss: 7.6364 - auc: 0.2146 - val_loss: 7.1004 - val_auc: 0.2047
Epoch 37/128
3/3 [==============================] - 0s 166ms/step - loss: 7.2766 - auc: 0.2215 - val_loss: 6.7612 - val_auc: 0.2103
Epoch 38/128
3/3 [==============================] - 0s 170ms/step - loss: 7.0067 - auc: 0.2267 - val_loss: 6.4557 - val_auc: 0.2422
Epoch 39/128
3/3 [==============================] - 0s 165ms/step - loss: 6.6749 - auc: 0.2504 - val_loss: 6.1393 - val_auc: 0.2670
Epoch 40/128
3/3 [==============================] - 0s 167ms/step - loss: 6.4055 - auc: 0.2752 - val_loss: 5.8013 - val_auc: 0.2730
Epoch 41/128
3/3 [==============================] - 0s 164ms/step - loss: 6.0406 - auc: 0.2853 - val_loss: 5.4472 - val_auc: 0.2763
Epoch 42/128
3/3 [==============================] - 0s 163ms/step - loss: 5.6900 - auc: 0.2959 - val_loss: 5.0857 - val_auc: 0.2753
Epoch 43/128
3/3 [==============================] - 0s 169ms/step - loss: 5.3493 - auc: 0.3021 - val_loss: 4.7238 - val_auc: 0.2759
Epoch 44/128
3/3 [==============================] - 0s 168ms/step - loss: 5.0003 - auc: 0.3173 - val_loss: 4.3651 - val_auc: 0.2831
Epoch 45/128
3/3 [==============================] - 0s 168ms/step - loss: 4.6351 - auc: 0.3360 - val_loss: 4.0155 - val_auc: 0.3100
Epoch 46/128
3/3 [==============================] - 0s 161ms/step - loss: 4.2634 - auc: 0.3625 - val_loss: 3.6813 - val_auc: 0.3567
Epoch 47/128
3/3 [==============================] - 0s 165ms/step - loss: 3.9849 - auc: 0.3852 - val_loss: 3.3769 - val_auc: 0.4130
Epoch 48/128
3/3 [==============================] - 0s 171ms/step - loss: 3.6751 - auc: 0.4363 - val_loss: 3.1234 - val_auc: 0.4702
Epoch 49/128
3/3 [==============================] - 0s 165ms/step - loss: 3.5281 - auc: 0.4986 - val_loss: 2.9567 - val_auc: 0.6022
Epoch 50/128
3/3 [==============================] - 0s 169ms/step - loss: 3.3288 - auc: 0.5729 - val_loss: 2.8431 - val_auc: 0.6643
Epoch 51/128
3/3 [==============================] - 0s 169ms/step - loss: 3.1976 - auc: 0.6140 - val_loss: 2.7664 - val_auc: 0.6802
Epoch 52/128
3/3 [==============================] - 0s 171ms/step - loss: 3.1641 - auc: 0.6307 - val_loss: 2.7063 - val_auc: 0.6868
Epoch 53/128
3/3 [==============================] - 0s 168ms/step - loss: 3.1325 - auc: 0.6367 - val_loss: 2.6482 - val_auc: 0.6902
Epoch 54/128
3/3 [==============================] - 0s 166ms/step - loss: 3.0224 - auc: 0.6475 - val_loss: 2.5867 - val_auc: 0.6920
Epoch 55/128
3/3 [==============================] - 0s 180ms/step - loss: 2.9425 - auc: 0.6490 - val_loss: 2.5214 - val_auc: 0.6923
Epoch 56/128
3/3 [==============================] - 0s 177ms/step - loss: 2.8656 - auc: 0.6506 - val_loss: 2.4545 - val_auc: 0.6925
Epoch 57/128
3/3 [==============================] - 0s 174ms/step - loss: 2.8168 - auc: 0.6441 - val_loss: 2.3884 - val_auc: 0.6917
Epoch 58/128
3/3 [==============================] - 0s 187ms/step - loss: 2.7198 - auc: 0.6452 - val_loss: 2.3251 - val_auc: 0.6912
Epoch 59/128
3/3 [==============================] - 0s 183ms/step - loss: 2.6257 - auc: 0.6462 - val_loss: 2.2642 - val_auc: 0.6905
Epoch 60/128
3/3 [==============================] - 0s 173ms/step - loss: 2.5911 - auc: 0.6405 - val_loss: 2.2035 - val_auc: 0.6906
Epoch 61/128
3/3 [==============================] - 0s 162ms/step - loss: 2.4895 - auc: 0.6436 - val_loss: 2.1434 - val_auc: 0.6915
Epoch 62/128
3/3 [==============================] - 0s 166ms/step - loss: 2.3829 - auc: 0.6517 - val_loss: 2.0830 - val_auc: 0.6931
Epoch 63/128
3/3 [==============================] - 0s 174ms/step - loss: 2.3260 - auc: 0.6587 - val_loss: 2.0216 - val_auc: 0.7022
Epoch 64/128
3/3 [==============================] - 0s 169ms/step - loss: 2.2563 - auc: 0.6661 - val_loss: 1.9584 - val_auc: 0.7124
Epoch 65/128
3/3 [==============================] - 0s 175ms/step - loss: 2.1475 - auc: 0.6781 - val_loss: 1.8916 - val_auc: 0.7240
Epoch 66/128
3/3 [==============================] - 0s 172ms/step - loss: 2.1068 - auc: 0.6845 - val_loss: 1.8222 - val_auc: 0.7370
Epoch 67/128
3/3 [==============================] - 0s 170ms/step - loss: 2.0274 - auc: 0.6965 - val_loss: 1.7505 - val_auc: 0.7467
Epoch 68/128
3/3 [==============================] - 0s 169ms/step - loss: 1.9410 - auc: 0.7034 - val_loss: 1.6764 - val_auc: 0.7622
Epoch 69/128
3/3 [==============================] - 0s 165ms/step - loss: 1.8377 - auc: 0.7182 - val_loss: 1.6001 - val_auc: 0.7729
Epoch 70/128
3/3 [==============================] - 0s 167ms/step - loss: 1.7602 - auc: 0.7233 - val_loss: 1.5217 - val_auc: 0.7825
Epoch 71/128
```

```
3/3 [==============================] - 0s 167ms/step - loss: 1.6528 - auc: 0.7374 - val_loss: 1.4416 - val_auc: 0.7889
Epoch 72/128
3/3 [==============================] - 0s 166ms/step - loss: 1.5905 - auc: 0.7348 - val_loss: 1.3604 - val_auc: 0.7941
Epoch 73/128
3/3 [==============================] - 0s 164ms/step - loss: 1.4793 - auc: 0.7449 - val_loss: 1.2793 - val_auc: 0.7982
Epoch 74/128
3/3 [==============================] - 0s 168ms/step - loss: 1.3701 - auc: 0.7558 - val_loss: 1.1997 - val_auc: 0.8011
Epoch 75/128
3/3 [==============================] - 0s 165ms/step - loss: 1.2730 - auc: 0.7657 - val_loss: 1.1235 - val_auc: 0.8029
Epoch 76/128
3/3 [==============================] - 0s 168ms/step - loss: 1.1960 - auc: 0.7722 - val_loss: 1.0536 - val_auc: 0.8065
Epoch 77/128
3/3 [==============================] - 0s 167ms/step - loss: 1.1171 - auc: 0.7837 - val_loss: 0.9923 - val_auc: 0.8182
Epoch 78/128
3/3 [==============================] - 0s 164ms/step - loss: 1.0550 - auc: 0.7993 - val_loss: 0.9418 - val_auc: 0.8450
Epoch 79/128
3/3 [==============================] - 0s 166ms/step - loss: 0.9699 - auc: 0.8308 - val_loss: 0.9021 - val_auc: 0.8734
Epoch 80/128
3/3 [==============================] - 0s 166ms/step - loss: 0.9360 - auc: 0.8458 - val_loss: 0.8717 - val_auc: 0.8877
Epoch 81/128
3/3 [==============================] - 0s 166ms/step - loss: 0.8924 - auc: 0.8613 - val_loss: 0.8512 - val_auc: 0.8893
Epoch 82/128
3/3 [==============================] - 0s 167ms/step - loss: 0.8862 - auc: 0.8682 - val_loss: 0.8385 - val_auc: 0.8892
Epoch 83/128
3/3 [==============================] - 0s 168ms/step - loss: 0.8683 - auc: 0.8725 - val_loss: 0.8309 - val_auc: 0.8889
Epoch 84/128
3/3 [==============================] - 0s 174ms/step - loss: 0.8604 - auc: 0.8732 - val_loss: 0.8264 - val_auc: 0.8892
Epoch 85/128
3/3 [==============================] - 0s 167ms/step - loss: 0.8508 - auc: 0.8793 - val_loss: 0.8232 - val_auc: 0.8894
Epoch 86/128
3/3 [==============================] - 0s 165ms/step - loss: 0.8439 - auc: 0.8811 - val_loss: 0.8208 - val_auc: 0.8899
Epoch 87/128
3/3 [==============================] - 0s 166ms/step - loss: 0.8587 - auc: 0.8769 - val_loss: 0.8187 - val_auc: 0.8911
Epoch 88/128
3/3 [==============================] - 0s 165ms/step - loss: 0.8413 - auc: 0.8794 - val_loss: 0.8164 - val_auc: 0.8916
Epoch 89/128
3/3 [==============================] - 0s 167ms/step - loss: 0.8378 - auc: 0.8823 - val_loss: 0.8138 - val_auc: 0.8924
Epoch 90/128
3/3 [==============================] - 0s 191ms/step - loss: 0.8259 - auc: 0.8862 - val_loss: 0.8106 - val_auc: 0.8926
Epoch 91/128
3/3 [==============================] - 0s 193ms/step - loss: 0.8196 - auc: 0.8869 - val_loss: 0.8070 - val_auc: 0.8931
Epoch 92/128
3/3 [==============================] - 0s 189ms/step - loss: 0.7947 - auc: 0.8945 - val_loss: 0.8034 - val_auc: 0.8932
Epoch 93/128
3/3 [==============================] - 0s 186ms/step - loss: 0.7978 - auc: 0.8933 - val_loss: 0.8000 - val_auc: 0.8932
Epoch 94/128
3/3 [==============================] - 0s 168ms/step - loss: 0.8003 - auc: 0.8906 - val_loss: 0.7925 - val_auc: 0.8936
Epoch 95/128
3/3 [==============================] - 0s 167ms/step - loss: 0.7915 - auc: 0.8935 - val_loss: 0.7912 - val_auc: 0.8942
Epoch 96/128
3/3 [==============================] - 0s 166ms/step - loss: 0.7766 - auc: 0.8971 - val_loss: 0.7904 - val_auc: 0.8946
Epoch 97/128
3/3 [==============================] - 0s 168ms/step - loss: 0.7759 - auc: 0.8989 - val_loss: 0.7883 - val_auc: 0.8948
Epoch 98/128
3/3 [==============================] - 0s 167ms/step - loss: 0.7645 - auc: 0.9004 - val_loss: 0.7840 - val_auc: 0.8950
Epoch 99/128
3/3 [==============================] - 0s 169ms/step - loss: 0.7463 - auc: 0.9046 - val_loss: 0.7812 - val_auc: 0.8954
Epoch 100/128
3/3 [==============================] - 0s 162ms/step - loss: 0.7545 - auc: 0.9041 - val_loss: 0.7816 - val_auc: 0.8954
Epoch 101/128
3/3 [==============================] - 0s 165ms/step - loss: 0.7442 - auc: 0.9053 - val_loss: 0.7818 - val_auc: 0.8954
Epoch 102/128
3/3 [==============================] - 0s 173ms/step - loss: 0.7477 - auc: 0.9054 - val_loss: 0.7795 - val_auc: 0.8956
Epoch 103/128
3/3 [==============================] - 0s 160ms/step - loss: 0.7260 - auc: 0.9097 - val_loss: 0.7773 - val_auc: 0.8960
Epoch 104/128
3/3 [==============================] - 0s 163ms/step - loss: 0.7239 - auc: 0.9114 - val_loss: 0.7768 - val_auc: 0.8960
Epoch 105/128
3/3 [==============================] - 0s 159ms/step - loss: 0.7072 - auc: 0.9142 - val_loss: 0.7783 - val_auc: 0.8958
Epoch 106/128
3/3 [==============================] - 0s 168ms/step - loss: 0.6901 - auc: 0.9189 - val_loss: 0.7779 - val_auc: 0.8959
Epoch 107/128
3/3 [==============================] - 0s 166ms/step - loss: 0.6960 - auc: 0.9173 - val_loss: 0.7776 - val_auc: 0.8959
Epoch 108/128
3/3 [==============================] - 0s 164ms/step - loss: 0.6912 - auc: 0.9181 - val_loss: 0.7769 - val_auc: 0.8961
Epoch 109/128
3/3 [==============================] - 0s 167ms/step - loss: 0.6783 - auc: 0.9219 - val_loss: 0.7770 - val_auc: 0.8959
Epoch 110/128
3/3 [==============================] - 0s 167ms/step - loss: 0.6716 - auc: 0.9231 - val_loss: 0.7766 - val_auc: 0.8962
Epoch 111/128
3/3 [==============================] - 0s 165ms/step - loss: 0.6815 - auc: 0.9214 - val_loss: 0.7766 - val_auc: 0.8962
Epoch 112/128
3/3 [==============================] - 0s 164ms/step - loss: 0.6633 - auc: 0.9239 - val_loss: 0.7787 - val_auc: 0.8956
Epoch 113/128
3/3 [==============================] - 0s 168ms/step - loss: 0.6572 - auc: 0.9258 - val_loss: 0.7794 - val_auc: 0.8955
Epoch 114/128
3/3 [==============================] - 0s 167ms/step - loss: 0.6536 - auc: 0.9267 - val_loss: 0.7779 - val_auc: 0.8962
Epoch 115/128
3/3 [==============================] - 0s 166ms/step - loss: 0.6454 - auc: 0.9288 - val_loss: 0.7782 - val_auc: 0.8962
Epoch 116/128
3/3 [==============================] - 0s 167ms/step - loss: 0.6444 - auc: 0.9291 - val_loss: 0.7817 - val_auc: 0.8952
Epoch 117/128
3/3 [==============================] - 0s 164ms/step - loss: 0.6399 - auc: 0.9310 - val_loss: 0.7837 - val_auc: 0.8948
Epoch 118/128
3/3 [==============================] - 0s 165ms/step - loss: 0.6227 - auc: 0.9335 - val_loss: 0.7845 - val_auc: 0.8947
Epoch 119/128
3/3 [==============================] - 0s 167ms/step - loss: 0.6217 - auc: 0.9352 - val_loss: 0.7834 - val_auc: 0.8955
Epoch 120/128
3/3 [==============================] - 0s 165ms/step - loss: 0.6172 - auc: 0.9346 - val_loss: 0.7848 - val_auc: 0.8953
Epoch 121/128
3/3 [==============================] - 0s 168ms/step - loss: 0.6056 - auc: 0.9362 - val_loss: 0.7898 - val_auc: 0.8939
Epoch 122/128
3/3 [==============================] - 0s 159ms/step - loss: 0.6204 - auc: 0.9346 - val_loss: 0.7893 - val_auc: 0.8944
Epoch 123/128
3/3 [==============================] - 0s 166ms/step - loss: 0.5952 - auc: 0.9391 - val_loss: 0.7896 - val_auc: 0.8948
Epoch 124/128
```

```
3/3 [==============================] - 0s 169ms/step - loss: 0.5889 - auc: 0.9405 - val_loss: 0.7927 - val_auc: 0.8939
Epoch 125/128
3/3 [==============================] - 0s 165ms/step - loss: 0.5754 - auc: 0.9431 - val_loss: 0.7959 - val_auc: 0.8933
Epoch 126/128
3/3 [==============================] - 0s 160ms/step - loss: 0.5876 - auc: 0.9412 - val_loss: 0.7982 - val_auc: 0.8928
Epoch 127/128
3/3 [==============================] - 0s 162ms/step - loss: 0.5791 - auc: 0.9425 - val_loss: 0.7998 - val_auc: 0.8929
Epoch 128/128
3/3 [==============================] - 0s 167ms/step - loss: 0.5614 - auc: 0.9458 - val_loss: 0.8013 - val_auc: 0.8930
```

In [106…

```
plot_history(history)
```



In [107…

```
# model1.save_weights(filepath='./tfidf_model/model2')
```

In [30]:

```
model1 = make_model(metrics=METRICS1,
                    input_shape=X_train.shape[1],
                    output_shape=y_train.shape[1],
                    bias_initializer=tf.keras.initializers.Zeros(),
                    kernel_initializer=keras.initializers.GlorotUniform(),
                    optimizer=keras.optimizers.Adam(learning_rate=0.001),
                    output_bias=None
                   )
model1.load_weights(filepath='./tfidf_model/model2')
```

Out[30]:

```
<tensorflow.python.training.tracking.util.CheckpointLoadStatus at 0x7fe073573d60>
```

In [40]:

```
model1_score = model1.evaluate(X_test, y_test, batch_size=1, verbose=0)
print('AUC Score : ', model1_score[1])
```

```
AUC Score :  0.8718721866607666
```

In [44]:

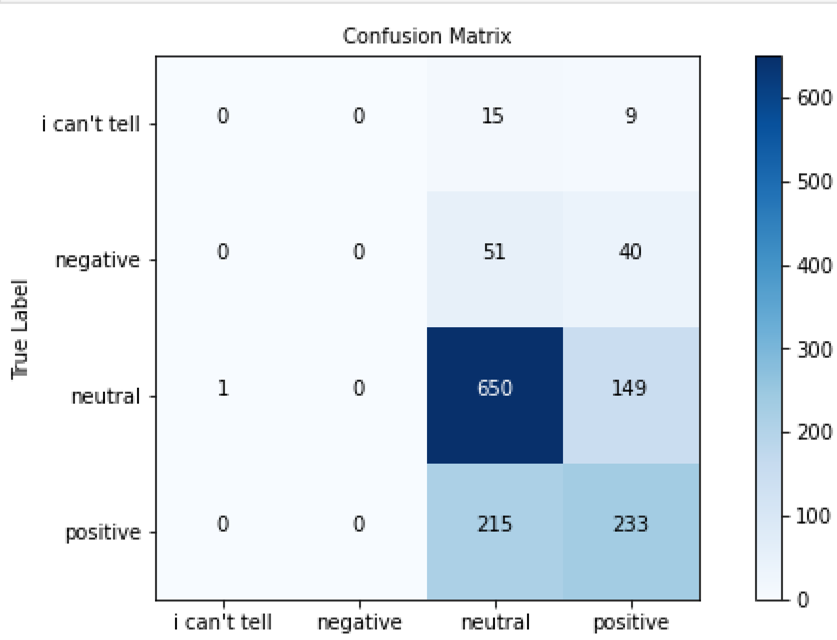
```
model1_y_pred = model1.predict(X_test, batch_size=1)
```

In [45]:

```
cm = confusion_matrix(y_test.argmax(axis=1), model1_y_pred.argmax(axis=1))
```

In [46]:

```
PlotConfusionMatrix(cm, ohe.categories_[0], FileName='./img/model1_cm.jpg')
```



Recurrent Neural Networks

In [49]:

```
l_encoder = LabelEncoder()
Y_true = l_encoder.fit_transform(df.sentiment)

tokenizer = Tokenizer(num_words=10000)
tokenizer.fit_on_texts(df.tweet)
sequences = tokenizer.texts_to_sequences(df.tweet)

word_index = tokenizer.word_index
data = pad_sequences(sequences, maxlen=100)
labels = to_categorical(np.asarray(Y_true))

print(f"{'Unique Words':25} {len(word_index)}")
print(f"{'Shape of data tensor':25} {data.shape}")
```

```
print(f"{'Shape of label tensor':25} {labels.shape}")
# print(labels)
```

```
Unique Words              7570
Shape of data tensor      (9092, 100)
Shape of label tensor     (9092, 4)
```

In [50]:

```
Test_Size = int(data.shape[0]*.2)

X_train1, X_test1, y_train1, y_test1 = train_test_split(data, labels, test_size=Test_Size, random_state=67)

X_train1, X_val1, y_train1, y_val1 = train_test_split(X_train1, y_train1, test_size=Test_Size, random_state=67)

print(f"Train\n\t{'X :':5}{X_train1.shape}\n\t{'Y :':5}{y_train1.shape}")
print(f"Test\n\t{'X :':5}{X_test1.shape}\n\t{'Y :':5}{y_test1.shape}")
print(f"Validation\n\t{'X :':5}{X_val1.shape}\n\t{'Y :':5}{y_val1.shape}")
```

```
Train
        X :  (5456, 100)
        Y :  (5456, 4)
Test
        X :  (1818, 100)
        Y :  (1818, 4)
Validation
        X :  (1818, 100)
        Y :  (1818, 4)
```

Since we have our train-validation split ready, our next step is to create an embedding matrix from the precomputed Glove embeddings

GloVe model "Twitter" pre-trained word vectors can be downloaded by the following link:

https://nlp.stanford.edu/projects/glove/

In [596…

```
#Generate a dictionary with glove vocabs as key and coefficients as numpy array
embeddings_index = dict()
twitter_27B_25d = open("./glove/glove.twitter.27B.25d.txt")
for line in twitter_27B_25d:
    values = line.split()
    vocab = values[0]
    coefs = np.asarray(values[1:], dtype='float32')
    embeddings_index[vocab] = coefs
twitter_27B_25d.close()
```

In [597…

```
# create a matrix of zeros with the size of of 'Word_index + 1 to match the input dimension of Embedding layer
oov_list = []
embedding_matrix = np.zeros((len(word_index) + 1, len(embeddings_index[next(iter(embeddings_index))])))
for word, idx in word_index.items():
    embedding_vector = embeddings_index.get(word)
    if embedding_vector is not None:
        embedding_matrix[idx] = embedding_vector
    else:
        oov_list.append(word)
with open('./glove/embedding_matrix.pkl', 'wb') as eFile:
    pickle.dump(embedding_matrix, eFile)
    eFile.close()
print('Embedding Matrix shape ', embedding_matrix.shape)
```

```
Embedding Matrix shape  (7573, 25)
```

In [51]:

```
with open('./glove/embedding_matrix.pkl', 'rb') as eFile:
    embedding_matrix = pickle.load(eFile)
```

In [52]:

```
InputDim = embedding_matrix.shape[0]
EmbeddinDim = embedding_matrix.shape[1]
MaxSequenceLenght = data.shape[1]
```

In [53]:

```
ClassWeights, SampleWeights = ohe_weights(y_train1)
bias = [x for x in ClassWeights.values()]
```

In [54]:

```
# filepath = './tfidf_model/model.{epoch:02d}-{val_loss:.2f}.h5'
filepath = './glove/model2.cptk'
CP_AUC=[
    ModelCheckpoint(filepath=filepath, monitor='val_auc',
                    save_best_only=False,save_weights_only=True,mode='max', save_freq=0),
    EarlyStopping(monitor='val_auc', min_delta=0.0001, patience=15,
                  verbose=1,mode='max',baseline=0.9999,restore_best_weights=False)
]
```

In [55]:

```
eStop_AUC       = EarlyStopping(monitor='auc', min_delta=0.0001, patience=15,
                                verbose=1,mode='max',baseline=0.9999,restore_best_weights=False)
eStop_Val_AUC   = EarlyStopping(monitor='val_auc', min_delta=0.001, patience=15,
                                verbose=1,mode='max',baseline=0.999,restore_best_weights=False)
eStop_TP        = EarlyStopping(monitor='TP', min_delta=0.0001, patience=24,
                               verbose=1,mode='max',baseline=0.999,restore_best_weights=False)
eStop_TN        = EarlyStopping(monitor='TN', min_delta=0.001, patience=24,
                               verbose=1,mode='max',baseline=0.999,restore_best_weights=False)
eStop_FP        = EarlyStopping(monitor='FP', min_delta=0.001, patience=24,
                               verbose=1,mode='min',baseline=0.999,restore_best_weights=False)
eStop_FN        = EarlyStopping(monitor='FN', min_delta=0.001, patience=24,
                               verbose=1,mode='min',baseline=0.999,restore_best_weights=False)
eStop_Reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.1, patience=50, min_lr=0.000001, verbose=1)
```

In [56]:

```
METRICS2 = [keras.metrics.AUC(name='auc')]
EPOCHS = 128
BATCH_SIZE = 128
STEPS_x_EPOCH = X_train1.shape[0]//BATCH_SIZE
```

Train a model with class weights

Now try re-training and evaluating the model with class weights to see how that affects the predictions.

In [60]:

```
keras.backend.clear_session()
```

In [61]:

```
model2=make_embedding_model(Metrics=METRICS,
                            Optimizer=keras.optimizers.Adam(),
                            Input_Dim=InputDim,
                            EmbeddinDim=EmbeddinDim,
                            Weights=embedding_matrix,
                            Input_Length=MaxSequenceLenght,
                            Output_Bias=None)
model2.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 100, 25)           189325
_________________________________________________________________
bidirectional (Bidirectional (None, 100, 50)           10200
_________________________________________________________________
lstm_1 (LSTM)                (None, 16)                4288
_________________________________________________________________
dense16 (Dense)              (None, 16)                272
_________________________________________________________________
dropout (Dropout)            (None, 16)                0
_________________________________________________________________
output4 (Dense)              (None, 4)                 68
=================================================================
Total params: 204,153
Trainable params: 14,828
Non-trainable params: 189,325
_________________________________________________________________
```

In [62]:

```
model2.layers[0].embeddings_initializer = keras.initializers.GlorotUniform()
model2.layers[-1].bias.assign([x for x in ClassWeights.values()])
model2.layers[3].activation = tf.keras.activations.swish
```

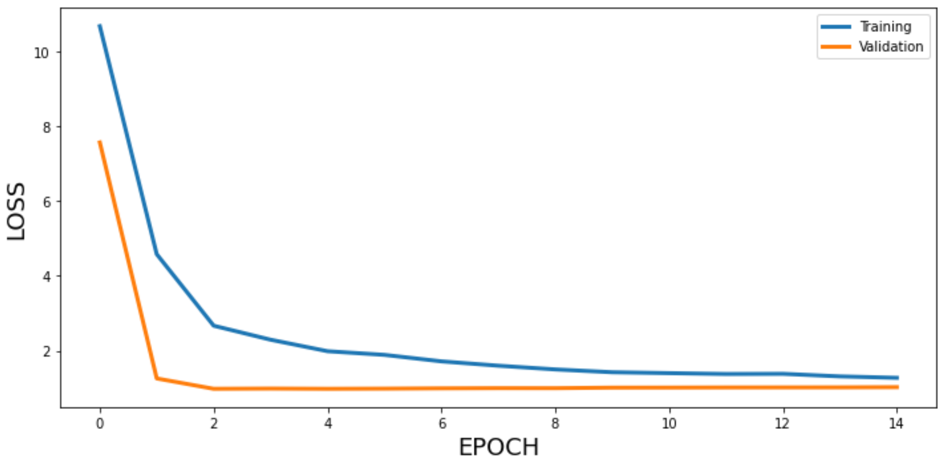
In [757…

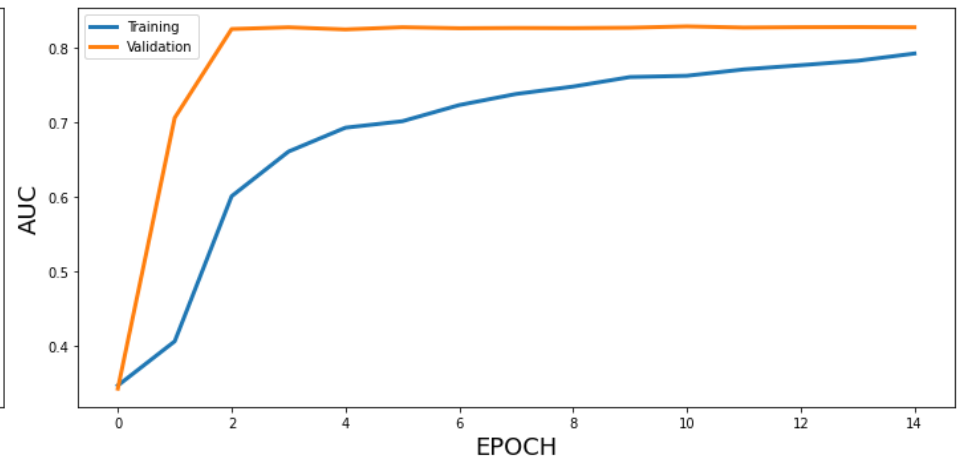
```
history = model2.fit(X_train1,
                     y_train1,
                     validation_data=(X_val1, y_val1),
                     validation_split=0.1,
                     validation_batch_size=1,
                     epochs=EPOCHS,
                     batch_size=BATCH_SIZE,
                     steps_per_epoch=STEPS_x_EPOCH,
                     callbacks=CP_AUC,
                     use_multiprocessing = True,
                     workers=6
                     )
```

```
Epoch 1/128
42/42 [==============================] - 14s 235ms/step - loss: 11.8230 - auc: 0.3449 - val_loss: 7.5637 - val_auc: 0.341
9
Epoch 2/128
42/42 [==============================] - 8s 197ms/step - loss: 5.8153 - auc: 0.3712 - val_loss: 1.2597 - val_auc: 0.7063
Epoch 3/128
42/42 [==============================] - 8s 192ms/step - loss: 2.8103 - auc: 0.5787 - val_loss: 0.9830 - val_auc: 0.8257
Epoch 4/128
42/42 [==============================] - 8s 192ms/step - loss: 2.3503 - auc: 0.6525 - val_loss: 0.9893 - val_auc: 0.8281
Epoch 5/128
42/42 [==============================] - 8s 192ms/step - loss: 2.0368 - auc: 0.6863 - val_loss: 0.9828 - val_auc: 0.8252
Epoch 6/128
42/42 [==============================] - 8s 199ms/step - loss: 1.9464 - auc: 0.6945 - val_loss: 0.9875 - val_auc: 0.8282
Epoch 7/128
42/42 [==============================] - 8s 195ms/step - loss: 1.7001 - auc: 0.7278 - val_loss: 0.9960 - val_auc: 0.8268
Epoch 8/128
42/42 [==============================] - 8s 198ms/step - loss: 1.6575 - auc: 0.7311 - val_loss: 1.0012 - val_auc: 0.8272
Epoch 9/128
42/42 [==============================] - 8s 199ms/step - loss: 1.5007 - auc: 0.7464 - val_loss: 1.0002 - val_auc: 0.8270
Epoch 10/128
42/42 [==============================] - 8s 191ms/step - loss: 1.3971 - auc: 0.7668 - val_loss: 1.0143 - val_auc: 0.8275
Epoch 11/128
42/42 [==============================] - 8s 192ms/step - loss: 1.3992 - auc: 0.7660 - val_loss: 1.0158 - val_auc: 0.8293
Epoch 12/128
42/42 [==============================] - 8s 194ms/step - loss: 1.3712 - auc: 0.7767 - val_loss: 1.0192 - val_auc: 0.8278
Epoch 13/128
42/42 [==============================] - 8s 195ms/step - loss: 1.3776 - auc: 0.7760 - val_loss: 1.0204 - val_auc: 0.8282
Epoch 14/128
42/42 [==============================] - 8s 192ms/step - loss: 1.3471 - auc: 0.7782 - val_loss: 1.0217 - val_auc: 0.8284
Epoch 15/128
42/42 [==============================] - 8s 198ms/step - loss: 1.3001 - auc: 0.7907 - val_loss: 1.0274 - val_auc: 0.8281
Epoch 00015: early stopping
```

In [764…

```
Plot_History(history)
```





In [766…]:

```
model2.save_weights('./glove/model2_auc')
```

In [74]:

```
keras.backend.clear_session()
```

In [57]:

```
METRICS2 = [keras.metrics.AUC(name='auc'), keras.metrics.TruePositives(name='TP'), keras.metrics.TrueNegatives(name='TN'
EPOCHS = 128
BATCH_SIZE = 128
STEPS_x_EPOCH = X_train1.shape[0]//BATCH_SIZE
```

In [58]:

```
model2=make_embedding_model(Metrics=METRICS2,
                            Optimizer=keras.optimizers.Adam(),
                            Input_Dim=InputDim,
                            EmbeddinDim=EmbeddinDim,
                            Weights=embedding_matrix,
                            Input_Length=MaxSequenceLenght,
                            Output_Bias=None)

model2.layers[0].embeddings_initializer = keras.initializers.GlorotUniform()
model2.layers[3].activation = tf.keras.activations.swish
model2.load_weights('./glove/model2_auc')
model2.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 100, 25)           189325
_________________________________________________________________
bidirectional (Bidirectional (None, 100, 50)           10200
_________________________________________________________________
lstm_1 (LSTM)                (None, 16)                4288
_________________________________________________________________
dense16 (Dense)              (None, 16)                272
_________________________________________________________________
dropout_2 (Dropout)          (None, 16)                0
_________________________________________________________________
output4 (Dense)              (None, 4)                 68
=================================================================
Total params: 204,153
Trainable params: 14,828
Non-trainable params: 189,325
_________________________________________________________________
```

In [43]:

```
history = model2.fit(X_train1,
                     y_train1,
                     validation_data=(X_val1, y_val1),
                     validation_split=0.1,
                     validation_batch_size=1,
                     epochs=EPOCHS,
                     batch_size=BATCH_SIZE,
                     steps_per_epoch=STEPS_x_EPOCH,
                     callbacks=[eStop_TP, eStop_TN],
                     use_multiprocessing = True,
                     workers=6
                     )
```

```
Epoch 1/128
42/42 [==============================] - 14s 226ms/step - loss: 1.3348 - auc: 0.7827 - TP: 1154.0233 - TN: 7372.3488 - va
l_loss: 1.0275 - val_auc: 0.8281 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 2/128
42/42 [==============================] - 7s 174ms/step - loss: 1.2542 - auc: 0.7954 - TP: 1237.1163 - TN: 7387.1628 - val
_loss: 1.0327 - val_auc: 0.8286 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 3/128
42/42 [==============================] - 7s 174ms/step - loss: 1.3724 - auc: 0.7879 - TP: 1245.7442 - TN: 7351.9070 - val
_loss: 1.0244 - val_auc: 0.8296 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 4/128
42/42 [==============================] - 7s 175ms/step - loss: 1.2593 - auc: 0.7958 - TP: 1225.3256 - TN: 7375.8140 - val
_loss: 1.0296 - val_auc: 0.8290 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 5/128
42/42 [==============================] - 7s 174ms/step - loss: 1.3407 - auc: 0.7974 - TP: 1254.5581 - TN: 7373.9535 - val
_loss: 1.0279 - val_auc: 0.8286 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 6/128
42/42 [==============================] - 7s 177ms/step - loss: 1.2386 - auc: 0.8049 - TP: 1289.4884 - TN: 7330.0000 - val
_loss: 1.0263 - val_auc: 0.8278 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 7/128
42/42 [==============================] - 7s 175ms/step - loss: 1.2915 - auc: 0.7909 - TP: 1256.2791 - TN: 7314.9302 - val
_loss: 1.0241 - val_auc: 0.8269 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 8/128
42/42 [==============================] - 7s 175ms/step - loss: 1.2269 - auc: 0.8027 - TP: 1269.9535 - TN: 7272.1163 - val
_loss: 1.0273 - val_auc: 0.8293 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 9/128
```

```
42/42 [==============================] - 7s 175ms/step - loss: 1.2187 - auc: 0.8010 - TP: 1279.0233 - TN: 7244.8140 - val
_loss: 1.0276 - val_auc: 0.8281 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 10/128
42/42 [==============================] - 7s 175ms/step - loss: 1.2671 - auc: 0.7916 - TP: 1231.2558 - TN: 7245.5581 - val
_loss: 1.0261 - val_auc: 0.8280 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 11/128
42/42 [==============================] - 7s 175ms/step - loss: 1.2449 - auc: 0.8012 - TP: 1288.5814 - TN: 7235.2093 - val
_loss: 1.0270 - val_auc: 0.8298 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 12/128
42/42 [==============================] - 7s 176ms/step - loss: 1.2271 - auc: 0.8001 - TP: 1229.2791 - TN: 7243.5349 - val
_loss: 1.0240 - val_auc: 0.8276 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 13/128
42/42 [==============================] - 7s 176ms/step - loss: 1.2310 - auc: 0.8041 - TP: 1313.1628 - TN: 7178.3953 - val
_loss: 1.0250 - val_auc: 0.8286 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 14/128
42/42 [==============================] - 7s 175ms/step - loss: 1.2595 - auc: 0.7923 - TP: 1284.9535 - TN: 7362.1860 - val
_loss: 1.0272 - val_auc: 0.8282 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 15/128
42/42 [==============================] - 7s 180ms/step - loss: 1.2797 - auc: 0.7972 - TP: 1294.7209 - TN: 7327.0233 - val
_loss: 1.0194 - val_auc: 0.8282 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 16/128
42/42 [==============================] - 8s 186ms/step - loss: 1.2263 - auc: 0.7978 - TP: 1299.8605 - TN: 7331.8140 - val
_loss: 1.0325 - val_auc: 0.8279 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 17/128
42/42 [==============================] - 8s 188ms/step - loss: 1.2728 - auc: 0.8000 - TP: 1332.5349 - TN: 7308.1860 - val
_loss: 1.0239 - val_auc: 0.8291 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 18/128
42/42 [==============================] - 7s 178ms/step - loss: 1.2211 - auc: 0.8052 - TP: 1281.6047 - TN: 7310.6279 - val
_loss: 1.0175 - val_auc: 0.8294 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 19/128
42/42 [==============================] - 7s 176ms/step - loss: 1.2894 - auc: 0.7944 - TP: 1303.0233 - TN: 7292.9070 - val
_loss: 1.0221 - val_auc: 0.8291 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 20/128
42/42 [==============================] - 7s 177ms/step - loss: 1.2729 - auc: 0.7917 - TP: 1291.2326 - TN: 7246.2558 - val
_loss: 1.0200 - val_auc: 0.8302 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 21/128
42/42 [==============================] - 7s 177ms/step - loss: 1.2615 - auc: 0.7954 - TP: 1270.7674 - TN: 7236.5116 - val
_loss: 1.0245 - val_auc: 0.8287 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 22/128
42/42 [==============================] - 7s 176ms/step - loss: 1.2639 - auc: 0.7975 - TP: 1342.1628 - TN: 7211.3488 - val
_loss: 1.0217 - val_auc: 0.8286 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 23/128
42/42 [==============================] - 7s 177ms/step - loss: 1.1899 - auc: 0.7983 - TP: 1286.4419 - TN: 7208.9535 - val
_loss: 1.0267 - val_auc: 0.8299 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 24/128
42/42 [==============================] - 7s 177ms/step - loss: 1.2670 - auc: 0.7999 - TP: 1308.8140 - TN: 7211.9070 - val
_loss: 1.0187 - val_auc: 0.8294 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 25/128
42/42 [==============================] - 7s 176ms/step - loss: 1.2145 - auc: 0.8086 - TP: 1324.7907 - TN: 7170.6512 - val
_loss: 1.0203 - val_auc: 0.8290 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 26/128
42/42 [==============================] - 7s 176ms/step - loss: 1.1692 - auc: 0.8075 - TP: 1336.4651 - TN: 7169.7674 - val
_loss: 1.0185 - val_auc: 0.8301 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 27/128
42/42 [==============================] - 7s 177ms/step - loss: 1.2377 - auc: 0.7969 - TP: 1336.5349 - TN: 7345.1860 - val
_loss: 1.0193 - val_auc: 0.8307 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 28/128
42/42 [==============================] - 7s 177ms/step - loss: 1.2163 - auc: 0.7983 - TP: 1354.6744 - TN: 7295.3023 - val
_loss: 1.0154 - val_auc: 0.8285 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 29/128
42/42 [==============================] - 7s 176ms/step - loss: 1.2276 - auc: 0.8001 - TP: 1341.1628 - TN: 7309.6744 - val
_loss: 1.0192 - val_auc: 0.8296 - val_TP: 322.0000 - val_TN: 1414.0000
Epoch 00029: early stopping
```

In [59]:

```
model2_score = model2.evaluate(X_test1, y_test1, batch_size=1, verbose=0)
print('AUC Accuracy :', model2_score[1])
```

```
AUC Accuracy : 0.8272119164466858
```

In [60]:

```
model2_y_pred = model2.predict(X_test1, batch_size=1, verbose=1)
```

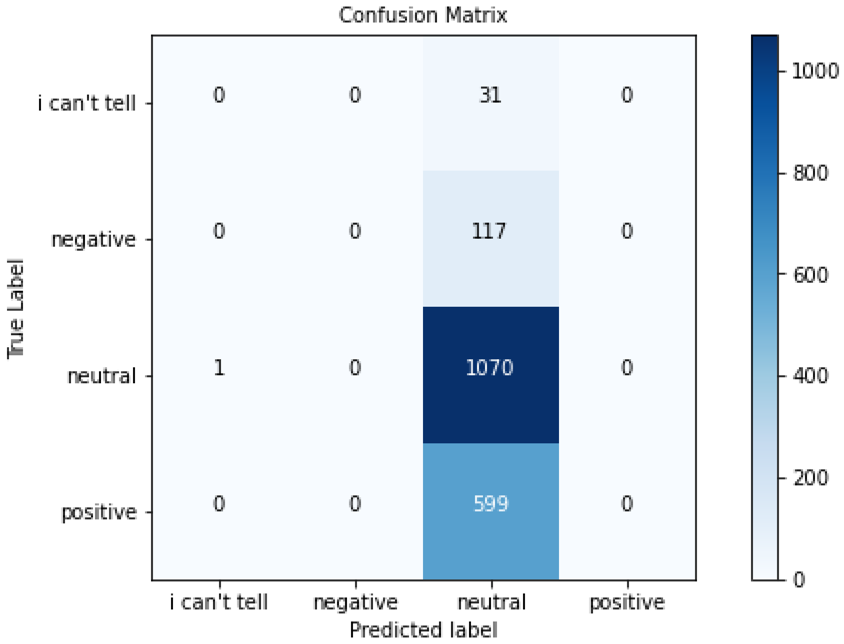
```
1818/1818 [==============================] - 15s 8ms/step
```

In [61]:

```
cm_model2 = confusion_matrix(y_test1.argmax(axis=1), model2_y_pred.argmax(axis=1))
```

In [62]:

```
PlotConfusionMatrix(cm_model2, ohe.categories_[0], FileName='./img/model2_cm.jpg')
```



Interpretation

- For the first model we used a simple Deep Neural Network with TF-IDF Vectorizer which results 87% of AUC accuracy but didn't able to predict more than 2 classes.

- The GloVe Embedding model constructed with Recurrent Neural Networks got less performance with 82% AUC accuracy.

- the overall result is not so much bad, dealing with a highly imbalanced dataset where the classe are respectively represented by 59.26%, 32.75% 6.27% and 1.72%, ML models wouldn't be able to learn. essentially we would need to have sufficiently large data to get better results.

Conclusion

- For the first model we used a simple Deep Neural Network with TF-IDF Vectorizer which results 87% of AUC accuracy but didn't able to predict more than 2 classes.
- The GloVe Embedding model constructed with Recurrent Neural Networks got less performance with 82% AUC accuracy.The overall result is not so much bad, dealing with a highly.
- The overall result is not so much bad, dealing with a highly. imbalanced dataset where the class distributions are respectively represented by 59.26%, 32.75% 6.27% and 1.72%, ML models wouldn't be able to learn. essentially we would need to have sufficiently large data to get better results.

Future Work

Data Collection

- Gather more data covering different social platforms from various sources like Facebook, Twitter, Instagram, YouTube etc..

Improve technical analysis by incorporating new features like

- Number of mentions
- Brand mentions
- Users geographic area
- Types and number of interactions