MACHINE LEARNING PROJECT

COURSE CODE-INT-254

Submitted by:

Shreyas Ninawe

12012928

A20

TOPIC: Social Media Sentimental Analysis Using Twitter Analysis

Dataset:
kaggle.com/code/muhammadimran112233/eda-twitter-sentiment-analysis-using-nn/notebook

Introduction

Millions of people are using Twitter and expressing their emotions like happiness, sadness, angry, etc. The Sentiment analysis is also about detecting the emotions, opinion, assessment, attitudes, and took this into consideration as a way humans think. Sentiment analysis classifies the emotions into classes such as positive or negative. Nowadays, industries are interested to use textual data for semantic analysis to extract the view of people about their products and services. Sentiment analysis is very important for them to know the customer satisfaction level and they can improve their services accordingly. To work on the text data, they try to extract the data from social media platforms. There are a lot of social media sites like Google Plus, Facebook, and Twitter that allow expressing opinions, views, and emotions about certain topics and events. Microblogging site Twitter is expanding rapidly among all other online social media networking sites with about 200 million users. Twitter was founded in 2006 and currently, it is the most famous microblogging platform. In 2017 2 million users shared 8.3 million tweets in one hour. Twitter users use to post their thoughts, emotions, and messages on their profiles, called tweets. Words limit of a single tweet has 140 characters. Twitter sentiment analysis based on the NLP (natural language processing) field. For tweets text, we use NLP techniques like tokenizing the words, removing the stop words like I, me, my, our, your, is, was, etc. Natural language processing also plays a part to preprocess the data like cleaning the text and removing the special characters and punctuation marks. Sentimental analysis is very important because we can know the trends of people's emotions on specific topics with their tweets.

Problem description/definition:

- To devise a sentimental analyzer for overcoming the challenges to identify the twitter tweets text sentiments (positive, negative) by implementing neural network using tensorflow

---

Evolution measures:

After training the model, we apply the evaluation measures to check that how the model is getting predictions. We will use the following evaluation measures to evaluate the performance of the models:

- Accuracy
- Confusion matrix with plot
- ROC Curve

Technical Approach

We are using python language in the implementations and Jupter Notebook that support the machine learning and data science projects. We will build tensorflow based model. We will use Sentiment 140 dataset and split that data into 70% for training and 30% for the testing purposes. After training on the model, we will evaluate the model to evaluate the performance of trained model

---

Source of Data:

---

https://www.kaggle.com/kazanova/sentime

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
plt.style.use('ggplot')
from sklearn.metrics import roc_curve, auc
from sklearn.metrics import classification_report, confusion_matrix
import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from sklearn.model_selection import train_test_split
from mlxtend.plotting import plot_confusion_matrix
import matplotlib.cm as cm
from matplotlib import rcParams
from collections import Counter
from nltk.tokenize import RegexpTokenizer
import re
import string
from tensorflow.keras.layers import LSTM, Activation, Dense, Dropout, Input, Embedding
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing import sequence
%matplotlib inline

import warnings
warnings.filterwarnings("ignore")
```

nt140

Loading the data 📁 📂

```
data = pd.read_csv("/kaggle/input/sentiment140/training.1600000.processed.noemoticon.csv", enco
ding = "ISO-8859-1", engine="python")
data.columns = ["label", "time", "date", "query", "username", "text"]
```

Exploratory data analysis 🔎 📊

Five top records of data

```
data.head()
```

	label	time	date	query	username	text
0	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
1	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
2	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
3	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....
4	0	1467811372	Mon Apr 06 22:20:00 PDT 2009	NO_QUERY	joy_wolf	@Kwesidei not the whole crew

In [3]:
```
data.head()
```

Out[3]:

	label	time	date	query	username	text
0	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
1	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
2	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
3	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....
4	0	1467811372	Mon Apr 06 22:20:00 PDT 2009	NO_QUERY	joy_wolf	@Kwesidei not the whole crew

Five last records of data

In [4]:
```
data.tail()
```

Out[4]:

	label	time	date	query	username	text
1599994	4	2193601966	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	AmandaMarie1028	Just woke up. Having no school is the best fee...
1599995	4	2193601969	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	TheWDBoards	TheWDB.com - Very cool to hear old Walt interv...
1599996	4	2193601991	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	bpbabe	Are you ready for your MoJo Makeover? Ask me f...
1599997	4	2193602064	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	tinydiamondz	Happy 38th Birthday to my boo of alll time!!! ...
1599998	4	2193602129	Tue Jun 16 08:40:50 PDT 2009	NO_QUERY	RyanTrevMorris	happy #charitytuesday @theNSPCC @SparksCharity...

In [8]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599999 entries, 0 to 1599998
Data columns (total 6 columns):
 #   Column    Non-Null Count    Dtype
---  ------    --------------    -----
 0   label     1599999 non-null  int64
 1   time      1599999 non-null  int64
 2   date      1599999 non-null  object
 3   query     1599999 non-null  object
 4   username  1599999 non-null  object
 5   text      1599999 non-null  object
dtypes: int64(2), object(4)
memory usage: 73.2+ MB
```

Data types of all coloumns

In [9]:
```
data.dtypes
```

Out[9]:
```
label        int64
time         int64
date        object
query       object
username    object
text        object
dtype: object
```

Data Preparation 

Model compilation

linkcode

- First we are calling the model
- We are using 2 classes so we set "binary_crossentropy" and if we use more than two classes then we use "categorical_crossentropy"
- Optimizer is a function that used to change the features of neural network such as learning rate (how the model learn with features) in order to reduce the losses. So the learning rate of neural network to reduce the losses is defined by optimizer.
- We are setting metrics=accuracy because we are going to caluclate the percentage of correct predictions over all predictions on the validation set

In [44]:
```
model = tensorflow_based_model() # here we are calling the function of created model
model.compile(loss='binary_crossentropy',optimizer=RMSprop(),metrics=['accuracy'])
```

Training and validating with parameter tuning

- We are feeding the training data and getting 10% data for validation from training data
- We set the following parameters:

- Batch size =80 so the model take 80 tweets in each iteration and train them. Batch size is a term used in machine learning and refers to the number of training examples utilized in one iteration.
- Epochs =6 so the model will train on the data 6 times.Epoch is a term used in machine learning and indicates the number of passes of the entire training dataset the machine learning algorithm has completed.
- We can choose batch_size, and epochs as we want so the good practice is to set some values and train the model if the model will not give the good results we can change it and then try again for the training of the model. We can repeat this process many time untill we will not get the good results and this process called as parameter tuning.

In [45]:

```
history=model.fit(X_train,Y_train,batch_size=80,epochs=6, validation_split=0.1)# here we are st
arting the training of model by feeding the training data
print('Training finished !!')
```

```
Epoch 1/6
315/315 [==============================] - 132s 414ms/step - loss: 0.6216 - accuracy: 0.6395
- val_loss: 0.5252 - val_accuracy: 0.7357
Epoch 2/6
315/315 [==============================] - 134s 424ms/step - loss: 0.4991 - accuracy: 0.7603
- val_loss: 0.5181 - val_accuracy: 0.7418
Epoch 3/6
315/315 [==============================] - 133s 422ms/step - loss: 0.4952 - accuracy: 0.7641
- val_loss: 0.5228 - val_accuracy: 0.7379
Epoch 4/6
315/315 [==============================] - 133s 421ms/step - loss: 0.4707 - accuracy: 0.7762
- val_loss: 0.5213 - val_accuracy: 0.7436
Epoch 5/6
315/315 [==============================] - 132s 420ms/step - loss: 0.4719 - accuracy: 0.7687
- val_loss: 0.5219 - val_accuracy: 0.7407
Epoch 6/6
315/315 [==============================] - 133s 423ms/step - loss: 0.4866 - accuracy: 0.7568
- val_loss: 0.5333 - val_accuracy: 0.7375
Training finished !!
```

We need to do all the above configurations to train the model. If we will not set all settings correctly then we could not get the desired results.

Testing the Trained model on test data

- Getting predictions/classifying the sentiments (positve/negative) on the test data using trained model.

In [46]:

```
accr1 = model.evaluate(X_test,Y_test) #we are starting to test the model here
```

```
375/375 [==============================] - 27s 71ms/step - loss: 0.5223 - accuracy: 0.7428
```

Accuracy

- Accuracy is the number of correctly classify tweets from all the tweets of positive and negative.
- For example, if the trained model classify the 70 tweets correct and 30 tweets wrong from total of 100 tweets then the accuracy score will be 70%.
- Accuracy= Total number of correct predictions/Total number of predictions

In [47]:

```
print('Test set\n  Accuracy: {:0.2f}'.format(accr1[1])) #the accuracy of the model on test data
is given below
```

```
Test set
  Accuracy: 0.74
```

Getting prediction of the test data and then we will compare the true labels/classes of the data with predictions

- As the model give probabilties so we are setting a threshold 0.5. More than 0.5 will be the positive tweets and lower will be negative tweets

In [48]:
```
y_pred = model.predict(X_test) #getting predictions on the trained model
y_pred = (y_pred > 0.5)
```

Confusion matrix

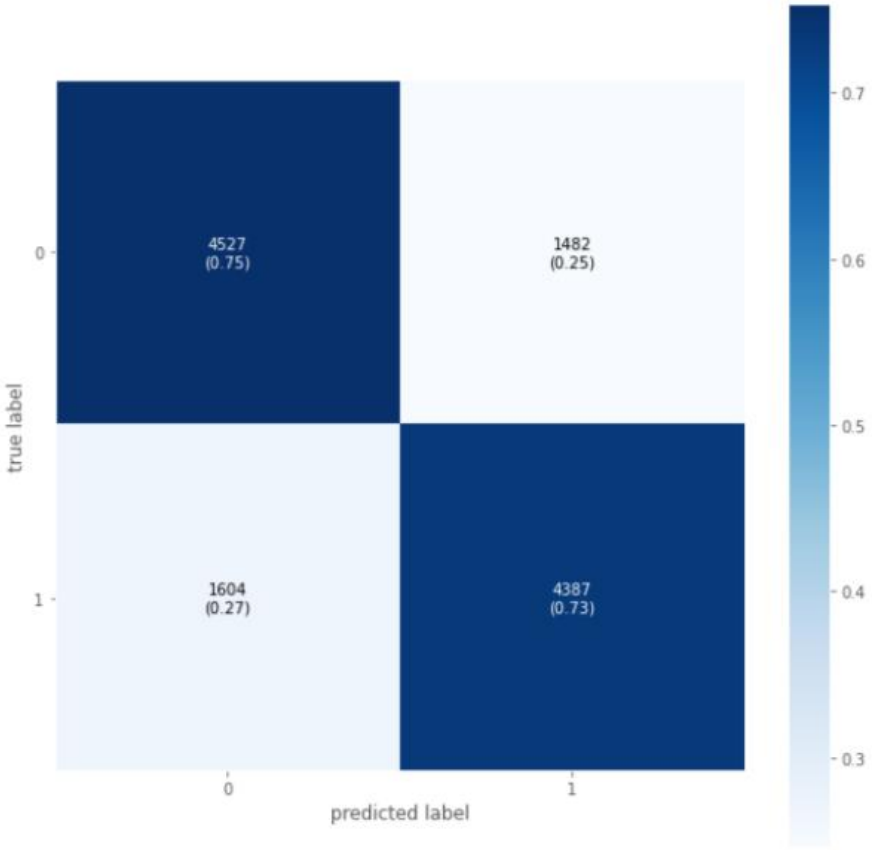
- These are the evaluation measures to evaluate the performance of the model.
- Dark blue boxes are the correct predictions with the trained model and sky blue boxes shows the wrong predictions.
- 4610 tweets correctly predicted as negative sentiments. 1399 tweets predicted positive sentiments but that were actually negative sentiments.
- 4247 tweets correctly predicted as postive sentiments. 1744 tweets predicted negative sentiments but that were actually positive sentiments.

In [49]:
```
print('\n')
print("confusion matrix")
print('\n')
CR=confusion_matrix(Y_test, y_pred)
print(CR)
print('\n')

fig, ax = plot_confusion_matrix(conf_mat=CR,figsize=(10, 10),
                                show_absolute=True,
                                show_normed=True,
                                colorbar=True)
plt.show()
```

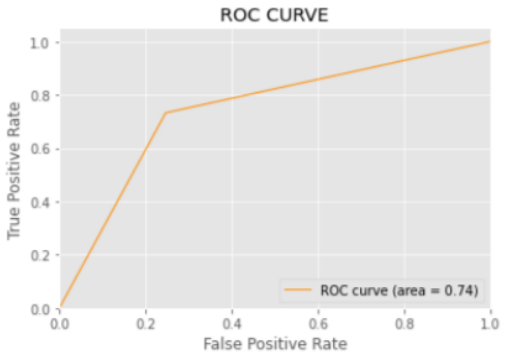
```
confusion matrix


[[4527 1482]
 [1604 4387]]
```



In [50]:

```
fpr, tpr, thresholds = roc_curve(Y_test, y_pred)
roc_auc = auc(fpr, tpr)
plt.figure()
plt.plot(fpr, tpr, color='darkorange', lw=1, label='ROC curve (area = %0.2f)' % roc_auc)
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC CURVE')
plt.legend(loc="lower right")
plt.show()
```



Conclusion

- We used the twitter sentiment analysis dataset and explored the data with different ways.
- We prepared the text data of tweets by removing the unnecessary things.
- We trained model based on tensorflow with all settings.
- We evaluated thye model with different evaluation measures.
- If you are interested to work on any text based project, you can simply apply the same methodolgy but might be you will need to change little settings like name of coloumns etc.
- We worked on the classification problem and sepcifically we call it binary classification which is two class classification.