Sentiment Analysis on Twitter for Positive and negative tweets

Python Code

Importing libraries

- import pandas as pd
- import matplotlib.pyplot as plt
- from sklearn.model_selection import train_test_split
- from sklearn.preprocessing import LabelEncoder
- from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
- from sklearn.manifold import TSNE
- from sklearn.feature_extraction.text import TfidfVectorizer
- from keras.preprocessing.text import Tokenizer
- from keras.preprocessing.sequence import pad_sequences
- from keras.models import Sequential
- from keras.layers import Activation, Dense, Dropout, Embedding, Flatten, Conv1D, MaxPooling1D, LSTM
- from keras import utils
- from keras.callbacks import ReduceLROnPlateau, EarlyStopping
- **# nltk**
- import nltk
- from nltk.corpus import stopwords
- from nltk.stem import SnowballStemmer
- # Word2vec
- import gensim
- # Utility
- import re
- import numpy as np
- import os
- from collections import Counter
- import logging
- import time
- import pickle
- import itertools
- # Set log
- logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.INFO)
- nltk.download('stopwords')

Data Sets :

0	1.468E+09	Mon Apr 06	NO_QUERY	_TheSpecia	@switchfoot http://twitpic.com/2y1zl - Awww, that's a bummer. You shoulda got David Carr of Third Day to do it. ;D
0	1.468E+09	Mon Apr 06	NO_QUERY	scotthamilt	is upset that he can't update his Facebook by texting it... and might cry as a result School today also. Blah!
0	1.468E+09	Mon Apr 06	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Managed to save 50% The rest go out of bounds
0	1.468E+09	Mon Apr 06	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
0	1.468E+09	Mon Apr 06	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all. i'm mad. why am i here? because I can't see you all over there.
0	1.468E+09	Mon Apr 06	NO_QUERY	joy_wolf	@Kwesidei not the whole crew
0	1.468E+09	Mon Apr 06	NO_QUERY	mybirch	Need a hug
0	1.468E+09	Mon Apr 06	NO_QUERY	coZZ	@LOLTrish hey long time no see! Yes.. Rains a bit ,only a bit LOL , I'm fine thanks , how's you ?
0	1.468E+09	Mon Apr 06	NO_QUERY	2Hood4Hol	@Tatiana_K nope they didn't have it
0	1.468E+09	Mon Apr 06	NO_QUERY	mimismo	@twittera que me muera ?
0	1.468E+09	Mon Apr 06	NO_QUERY	erinx3leanr	spring break in plain city... it's snowing
0	1.468E+09	Mon Apr 06	NO_QUERY	pardonlaur	I just re-pierced my ears
0	1.468E+09	Mon Apr 06	NO_QUERY	TLeC	@caregiving I couldn't bear to watch it. And I thought the UA loss was embarrassing . . . . .
0	1.468E+09	Mon Apr 06	NO_QUERY	robrobbier	@octolinz16 It it counts, idk why I did either. you never talk to me anymore
0	1.468E+09	Mon Apr 06	NO_QUERY	bayofwolve	@smarrison i would've been the first, but i didn't have a gun. not really though, zac snyder's just a doucheclown.
0	1.468E+09	Mon Apr 06	NO_QUERY	HairByJess	@iamjazzyfizzle I wish I got to watch it with you!! I miss you and @iamlilnicki how was the premiere?!
0	1.468E+09	Mon Apr 06	NO_QUERY	lovesongwi	Hollis' death scene will hurt me severely to watch on film wry is directors cut not out now?
0	1.468E+09	Mon Apr 06	NO_QUERY	armotley	about to file taxes
0	1.468E+09	Mon Apr 06	NO_QUERY	starkissed	@LettyA ahh ive always wanted to see rent love the soundtrack!!
0	1.468E+09	Mon Apr 06	NO_QUERY	gi_gi_bee	@FakerPattyPattz Oh dear. Were you drinking out of the forgotten table drinks?
0	1.468E+09	Mon Apr 06	NO_QUERY	quanvu	@alydesigns i was out most of the day so didn't get much done
0	1.468E+09	Mon Apr 06	NO_QUERY	swinspeed	one of my friend called me, and asked to meet with her at Mid Valley today...but i've no time *sigh*
0	1.468E+09	Mon Apr 06	NO_QUERY	cooliodoc	@angry_barista I baked you a cake but I ated it

- DATASET_COLUMNS = ["target", "ids", "date", "flag", "user", "text"]
- DATASET_ENCODING = "ISO-8859-1"
- TRAIN_SIZE = 0.8
- # TEXT CLENAING
- TEXT_CLEANING_RE = "@\S+|https?:\S+|http?:\S|[^A-Za-z0-9]+"
- # WORD2VEC
- W2V_SIZE = 300
- W2V_WINDOW = 7
- W2V_EPOCH = 32
- W2V_MIN_COUNT = 10
- # KERAS
- SEQUENCE_LENGTH = 300
- EPOCHS = 8
- BATCH_SIZE = 1024
- # SENTIMENT
- POSITIVE = "POSITIVE"
- NEGATIVE = "NEGATIVE"
- NEUTRAL = "NEUTRAL"
- SENTIMENT_THRESHOLDS = (0.4, 0.7)
- # EXPORT
- KERAS_MODEL = "model.h5"
- WORD2VEC_MODEL = "model.w2v"
- TOKENIZER_MODEL = "tokenizer.pkl"
- ENCODER_MODEL = "encoder.pkl"
- #path ="/"
- #dataset_filename = os.listdir(path)[0]
- #dataset_path = os.path.join("..",path,dataset_filename)
- #print("Open file:", dataset_path)
- #df = pd.read_csv(dataset_path, encoding =DATASET_ENCODING , names=DATASET_COLUMNS)
- df = pd.read_csv("D:/DataEntry/sdataset.csv", encoding =DATASET_ENCODING , names=DATASET_COLUMNS)

- print("Dataset size:", len(df))

Machine Learning Models:

1-Sentiment Analysis

- def decode_sentiment(label):
- return decode_map[int(label)]
- df.target = df.target.apply(lambda x: decode_sentiment(x))

- target_cnt = Counter(df.target)

- plt.figure(figsize=(16,8))
- plt.bar(target_cnt.keys(), target_cnt.values())
- plt.title("Dataset labels distribuition")

- stop_words = stopwords.words("english")
- stemmer = SnowballStemmer("english")
- def preprocess(text, stem=False):
- # Remove link,user and special characters
- text = re.sub(TEXT_CLEANING_RE, ' ', str(text).lower()).strip()
- tokens = []
- for token in text.split():
- if token not in stop_words:
- if stem:
- tokens.append(stemmer.stem(token))
- else:
- tokens.append(token)
- return " ".join(tokens)
- df.text = df.text.apply(lambda x: preprocess(x))

- df_train, df_test = train_test_split(df, test_size=1-TRAIN_SIZE, random_state=42)

- documents = [_text.split() for _text in df_train.text]

- w2v_model = gensim.models.word2vec.Word2Vec(size=W2V_SIZE,
- window=W2V_WINDOW,
- min_count=W2V_MIN_COUNT,
- workers=8)
- w2v_model.build_vocab(documents)

- words = w2v_model.wv.vocab.keys()
- vocab_size = len(words)
- w2v_model.train(documents, total_examples=len(documents), epochs=W2V_EPOCH)
- w2v_model.most_similar("love")
- tokenizer = Tokenizer()
- tokenizer.fit_on_texts(df_train.text)

- vocab_size = len(tokenizer.word_index) + 1
- x_train = pad_sequences(tokenizer.texts_to_sequences(df_train.text), maxlen=SEQUENCE_LENGTH)

- x_test = pad_sequences(tokenizer.texts_to_sequences(df_test.text), maxlen=SEQUENCE_LENGTH)
- labels = df_train.target.unique().tolist()
- labels.append(NEUTRAL)
- labels
- encoder = LabelEncoder()
- encoder.fit(df_train.target.tolist())

- y_train = encoder.transform(df_train.target.tolist())
- y_test = encoder.transform(df_test.target.tolist())

- y_train = y_train.reshape(-1,1)
- y_test = y_test.reshape(-1,1)

- y_train[:10]
- embedding_matrix = np.zeros((vocab_size, W2V_SIZE))
- for word, i in tokenizer.word_index.items():
- if word in w2v_model.wv:
- embedding_matrix[i] = w2v_model.wv[word]
- print(embedding_matrix.shape)
- embedding_layer = Embedding(vocab_size, W2V_SIZE, weights=[embedding_matrix], input_length=SEQUENCE_LENGTH, trainable=False)
- model = Sequential()
- model.add(embedding_layer)
- model.add(Dropout(0.5))
- model.add(LSTM(100, dropout=0.2, recurrent_dropout=0.2))
- model.add(Dense(1, activation='sigmoid'))

- model.summary()
- model.compile(loss='binary_crossentropy',
- optimizer="adam",
- metrics=['accuracy'])
- callbacks = [ReduceLROnPlateau(monitor='val_loss', patience=5, cooldown=0),
- EarlyStopping(monitor='val_acc', min_delta=1e-4, patience=5)]
- history = model.fit(x_train, y_train,
- batch_size=BATCH_SIZE,
- epochs=EPOCHS,
- validation_split=0.1,
- verbose=1,
- callbacks=callbacks)
- score = model.evaluate(x_test, y_test, batch_size=BATCH_SIZE)
- print()
- print("ACCURACY:",score[1])
- print("LOSS:",score[0])

- def decode_sentiment(score, include_neutral=True):
- if include_neutral:
- label = NEUTRAL
- if score <= SENTIMENT_THRESHOLDS[0]:
- label = NEGATIVE
- elif score >= SENTIMENT_THRESHOLDS[1]:
- label = POSITIVE

- return label
- else:
- return NEGATIVE if score < 0.5 else POSITIVE
- def predict(text, include_neutral=True):
- start_at = time.time()
- # Tokenize text
- x_test = pad_sequences(tokenizer.texts_to_sequences([text]), maxlen=SEQUENCE_LENGTH)
- # Predict
- score = model.predict([x_test])[0]
- # Decode sentiment
- label = decode_sentiment(score, include_neutral=include_neutral)

- return {"label": label, "score": float(score),
- "elapsed_time": time.time()-start_at}
- #Result =predict("I love the music")
- Result =predict("I hate the rain")
- print(Result)
- y_pred_1d = []
- y_test_1d = list(df_test.target)
- scores = model.predict(x_test, verbose=1, batch_size=8000)
- y_pred_1d = [decode_sentiment(score, include_neutral=False) for score in scores]

- accuracy_score(y_test_1d, y_pred_1d)
- model.save(KERAS_MODEL)
- w2v_model.save(WORD2VEC_MODEL)

Naïve Bayes

- print("naive bayes")

- from sklearn.naive_bayes import GaussianNB

- model = GaussianNB()
- DATASET_ENCODING = "ISO-8859-1"

- XDATASET_COLUMNS = ["text" , "Results"]
- dt = pd.read_csv("D:/DataEntry/Sampledata.csv", encoding =DATASET_ENCODING , names=XDATASET_COLUMNS)

- print(dt)

- X = dt.drop("Results" , axis=1)
- y= dt["Results"]

- labelencoder = LabelEncoder()
- X["text"] = labelencoder.fit_transform(X["text"])
- X_train , X_test ,Y_train , Y_test = train_test_split(X,y ,test_size=0.25 ,random_state=42)

- print(X_test)

- model = GaussianNB()

- model.fit(X_train, Y_train)

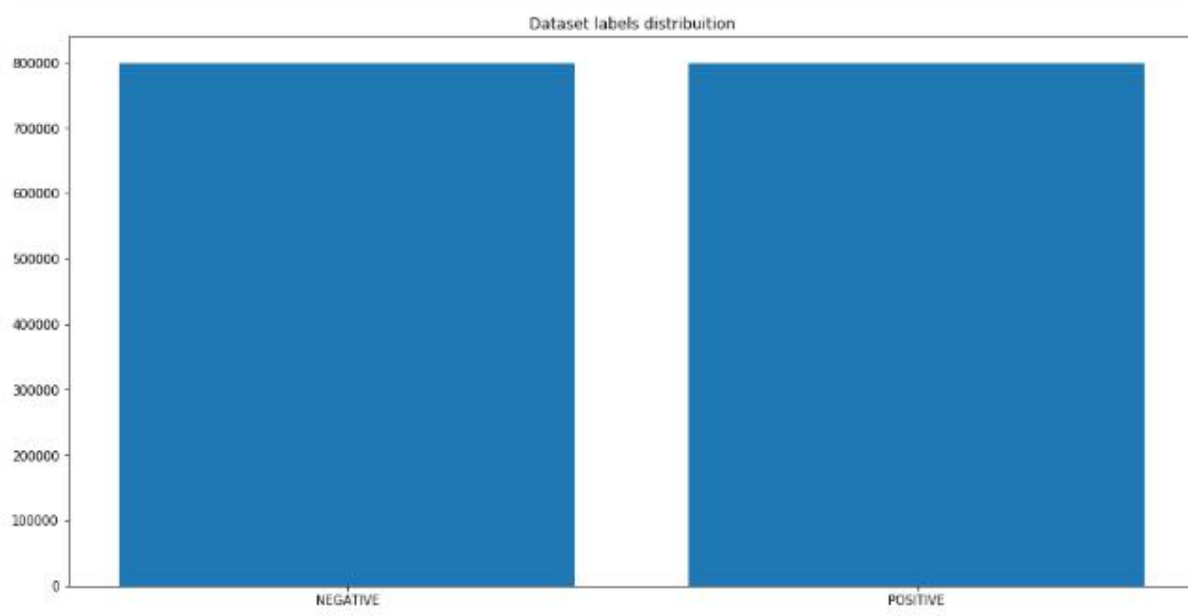
- y_pred = model.predict(X_test)

- df1 = pd.concat([X_test.reset_index(drop='Tru e'),y_pred.reset_index(drop='Tru e')],axis=1)
- # Train the model

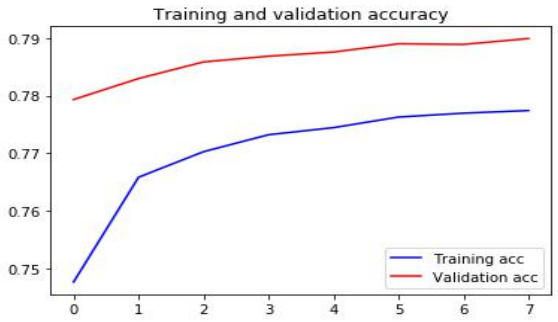
- #pickle.dump(tokenizer, open(TOKENIZER_MODEL, "wb"), protocol=0)
- #pickle.dump(encoder, open(ENCODER_MODEL, "wb"), protocol=0)

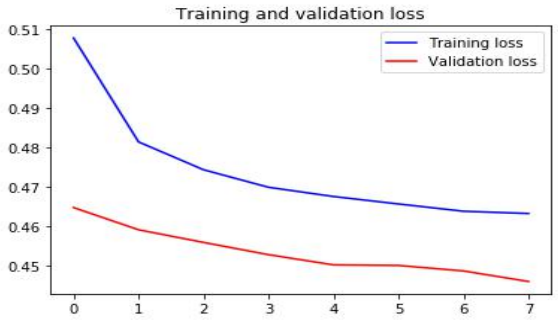
Visualizations:

Bar Chart:

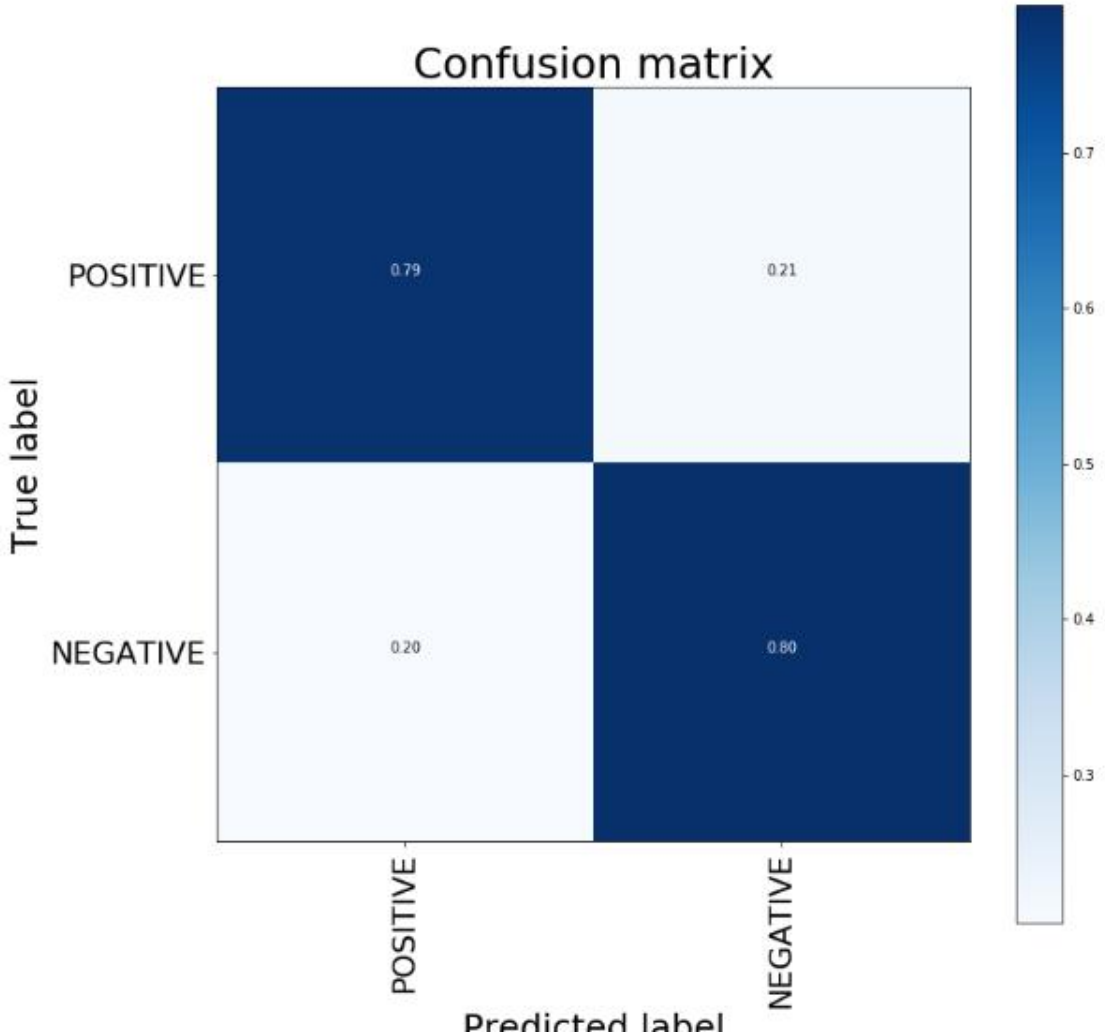


Epochs:





```
confusion_matrix
```



Output:

Predict :

Result =predict("I love the music")

Result:

Key	Type	Size	Value
elapsed_time	float	1	0.3728067874908447
label	str	1	NEUTRAL
score	float	1	0.43893977999687195