SOCIAL MEDIA ANALYTICS FOR BUSINESS INTELLIGENCE

DAT- 9731 – BOS1

INDIVIDUAL PROJECT

"FACEBOOK SENTIMENT ANALYSIS"

Under able guidance of –

PROF. BEAU GIANNINI & PROF. PAVEL PARAMONOV

Submitted by –

SHUBHAM MITTAL

FACEBOOK SENTIMENT ANALYSIS

Introduction

With the growing popularity of digital marketing and competition among brands, it is essential for companies to gain fair understanding of their customer's perceptions about the brands and products, especially on social media platforms like Facebook, with huge customer base on these social media platforms there is abundant data available based on the reviews given by customers towards brands and products they frequently use or have used as trial purchase. A Sentiment Analysis can be conducted to analyse comments made by customers as a feedback to brands and products, with the help of sentiment analysis companies can identify positive and negative comments on the social media platform, resolve issues pertaining to understanding of emotional tone and can get access to real time analysis to stay updated with trends.

Usage of Sentiment Analysis of Businesses

It is difficult to organize text based data, and also includes very complex and long analysing process. With majority of data being unorganized, it is essential to deploy a mechanism to analyse this data effectively and with maximum accuracy, which can be done using sentiment analysis. Sentiment Analysis has other advantages to the business which are as follows –

1. **Scalability –** the sentiment analysis provides a quick and cost-effective data processing for the companies to analyse the big data available on various social media platform in unstructured form.
2. **Real Time data analysis** – with the help of sentiment analysis companies can stay in touch with their customers in real time and get regular updates based on live information received from various social media platforms.
3. **Consistency –** It is essential to have a meaningful algorithm to analyse the emotional tone of comments made by customers and it can only be done 60 percent of time, with the help of sentiment analysis this can be done consistently with maximum accuracy.
4. **Perception Analysis –** it enables companies to identify the positive and negative comments made by customers on social media platforms, to access the brand salience and to engage with audience better.
5. **Influencer Searching –** it also provides insights to companies about the customers which are very happy with the brands and products, and have also shown maximum engagement with the brand.

Data Used in Analysis

1. A sample dataset has been downloaded from 'Kaggle' website which includes Facebook comments based on real time data extracted from the social media platform.
2. For the python code the file name "fb_sentiment.csv" has been used for analysis for facebook comments made by users, and python code has been executed to further simply the data as per requirement and accurate analysis.

Steps Followed in Analysis

1. Importing pandas, numpy and matplotlib.pyplot libraries to python environment, resulting in the access to all functions available in these libraries which can be used anytime in the program as per requirement.

```
In [58]: #Social Media Analytics for Business Intelligence - Individual Assignment (DAT - 9731, BOS1)

In [59]: import pandas as pd #importing library from pandas

In [60]: import numpy as np #importing numpy array

In [61]: import matplotlib.pyplot as plt #importing matplotlib
```

2. 'NLTK' is a powerful python package which provides a diverse set of algorithms, it is well documented and free to use for all. It consists of common algorithms such as tokenizing, sentiment analysis, stemming etc. Hence, this package has been installed. 'Tensorflow' is another python library for fast numerical computing created by Google, which would also enable many functions like Keras to enable fast computing of python code.

```
In [62]: pip install nltk #installting nltk
         Requirement already satisfied: nltk in c:\users\hp\anaconda3\lib\site-packages (3.6.1)
         Requirement already satisfied: joblib in c:\users\hp\anaconda3\lib\site-packages (from nltk) (1.0.1)
         Requirement already satisfied: click in c:\users\hp\anaconda3\lib\site-packages (from nltk) (7.1.2)
         Requirement already satisfied: tqdm in c:\users\hp\anaconda3\lib\site-packages (from nltk) (4.59.0)
         Requirement already satisfied: regex in c:\users\hp\anaconda3\lib\site-packages (from nltk) (2021.4.4)
         Note: you may need to restart the kernel to use updated packages.

In [63]: pip install tensorflow #installing tensorflow for keras
         Requirement already satisfied: tensorflow in c:\users\hp\anaconda3\lib\site-packages (2.5.0)
         Requirement already satisfied: wrapt~=1.12.1 in c:\users\hp\anaconda3\lib\site-packages (from tensorflow) (1.12.1)
         Requirement already satisfied: protobuf>=3.9.2 in c:\users\hp\anaconda3\lib\site-packages (from tensorflow) (3.17.3)
         Requirement already satisfied: tensorflow-estimator<2.6.0,>=2.5.0rc0 in c:\users\hp\anaconda3\lib\site-packages (from tensorflow)
         (2.5.0)
```

3. Using pd.read_csv command file 'fb_sentiment.csv' has been imported into the python environment, to enable access towards data available in the file.

```
In [64]: fb=pd.read_csv(r'C:\Users\hp\Desktop\Hult Boston\Social Media Analytics\fb_sentiment.csv') #reading the csv file
```

4. The data obtained from step 3 has been stored in variable 'fb', using 'fb.head()' command the first five rows have been identified to check the accuracy of data and identification of right rows and columns in the data set imported into the python environment.

```
In [65]: fb.head() #reading first five rows from table
```

Out[65]:

	Unnamed: 0	FBPost	Label
0	0	Drug Runners and a U.S. Senator have somethin...	O
1	1	Heres a single, to add, to Kindle. Just read t...	O
2	2	If you tire of Non-Fiction.. Check out http://...	O
3	3	Ghost of Round Island is supposedly nonfiction.	O
4	4	Why is Barnes and Nobles version of the Kindle...	N

5. Using command ‘fb.columns = map(str.lower, fb.columns)’, data obtained from the file ‘fb_sentiment.csv’ has been arranged in lower case for the column names. The data frame created using above mentioned command can be checked using command ‘fb.shape’ resulting in 1000 rows and 3 columns.

```
In [66]: # Lower-casing the coloumn names
         fb.columns=map(str.lower, fb.columns)

In [67]: # checkin the shape of the DF
         fb.shape

Out[67]: (1000, 3)
```

6. A regular expression or (re) function enables to check whether a particular string matches a regular expression or not. In the below syntax, ‘Lambda’ function has been used as we require a nameless function for a very short period of time. In python we can use ‘Lambda’ function as a temporary function. Hence, the syntax has been used to convert data related to ‘fbpost’ into lowercase and also to clean the data if there are any absurd or missing values.
   NLTK data package includes a pre-trained ‘punkt’ tokenizer for English language, which divides the text into list of sentences by using an algorithm to build a model for collocations and words which appear at the start of sentences.

```
In [68]: #Lowercasing the text and removing symbols though RegEx
         import re
         fb['fbpost'] = fb['fbpost'].apply(lambda x: x.lower())
         fb['fbpost'] = fb['fbpost'].apply((lambda x: re.sub('[^a-zA-z0-9\s]','',x)))
         fb = fb[fb.label != "O"]
         max_fatures = 2000

In [69]: import nltk.data
         nltk.download('punkt')

         [nltk_data] Downloading package punkt to
         [nltk_data]     C:\Users\hp\AppData\Roaming\nltk_data...
         [nltk_data]   Package punkt is already up-to-date!

Out[69]: True
```

7. For an effective sentiment analysis, it is essential to use tokenizer, which can split words from text to identify whether the comment is ‘positive’, ‘negative’ or ‘neutral’. This has been implemented by installing ‘mosestokenizer’ package in the python environment, which has further functions likes ‘mosesdetokenizer’. Mosestokenizer can normalize, punctuate, split sentences etc. for the entire data.

```
In [70]: pip install mosestokenizer

         Requirement already satisfied: mosestokenizer in c:\users\hp\anaconda3\lib\site-packages (1.1.0)
         Requirement already satisfied: toolwrapper in c:\users\hp\anaconda3\lib\site-packages (from mosestokenizer) (2.1.0)
         Requirement already satisfied: uctools in c:\users\hp\anaconda3\lib\site-packages (from mosestokenizer) (1.3.0)
         Requirement already satisfied: openfile in c:\users\hp\anaconda3\lib\site-packages (from mosestokenizer) (0.0.7)
         Requirement already satisfied: docopt in c:\users\hp\anaconda3\lib\site-packages (from mosestokenizer) (0.6.2)
         Note: you may need to restart the kernel to use updated packages.

In [71]: from mosestokenizer import MosesTokenizer, MosesDetokenizer
```

8. It is required to implement a neural network which can classify digits from an image in python environment using Tensorflow module. In addition to Tensorflow, another model which can used to design a neural network is ‘Keras’, which is less prone to errors than Tensorflow. Keras offers various APIs which can used to define a neural network as follows –

- Functional API – It is a full-featured API which supports arbitrary models, functional API is very flexible but complex than sequential API.

- Sequential API – It enables programmers to create models layer by layer for a defined problem statement. It is based on single-input and single-output stacks of layers and this model is very straightforward.

```
In [74]: from sklearn.feature_extraction.text import CountVectorizer
         from keras.preprocessing.text import Tokenizer
         from keras.preprocessing.sequence import pad_sequences
         from keras.models import Sequential
         from keras.layers import Dense, Embedding, LSTM, SpatialDropout1D
         from sklearn.model_selection import train_test_split
         from keras.utils.np_utils import to_categorical
         import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
```

9. With the help of above installed packages, following code can be used to design a neural network, based on sequential modelling using 'Keras' package. Before ingesting the data into the model, it is essential to use tokenizer to clean, split the text in the required format to remove data values which are not required. Using 'train' and 'test' command, the data is first trained for the model and then tested to get required output. The output

```
In [75]: import re
         fb['fbpost'] = fb['fbpost'].apply(lambda x: x.lower())
         fb['fbpost'] = fb['fbpost'].apply((lambda x: re.sub('[^a-zA-z0-9\s]','',x)))
         fb = fb[fb.label != "O"]
         max_fatures = 2000
         tokenizer = Tokenizer(num_words=max_fatures, split=' ')
         tokenizer.fit_on_texts(fb['fbpost'].values)
         X = tokenizer.texts_to_sequences(fb['fbpost'].values)
         X = pad_sequences(X)
         fb.label.value_counts()
         Y = pd.get_dummies(fb['label']).values
         X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.33, random_state = 42)
         #Here the lables are checked after the removal of the "other" sentimented comments.
         #Also some preperation to the algorithm, as preparing the test-, and training sets are done.
         print(X_train.shape,Y_train.shape)
         print(X_test.shape,Y_test.shape)
         #The Neural Network
         #In this section, the algorithm is prepared with following features:
         #The model is Sequential
         #The model type is an LSTM model
         embed_dim = 200
         lstm_out = 200

         (482, 203) (482, 2)
         (238, 203) (238, 2)
```

of the code has been obtained in form of a matrix using python code –
print (X_train.shape, Y_train.shape)

10. The following code has been designed and implemented to train the model as per the requirement, to produce the output in the form of a histogram. Using python code – print(model.summary()) a summary has been obtained, using code sequences with 'model.add' various features have been added to the model like length, shape, activations and compilation matrix. A histogram has been plotted with batch size of 32 and 7 epochs, along with data accuracy and loss variations of the data used for the sentiment analysis.
Using python code 'plt.show()' the histogram has been plotted. The model has been validated further to predict the 'positive', 'negative' score from the output obtained. The validation size used is 1500 in this case to enable the model access all the data points present in the data set.

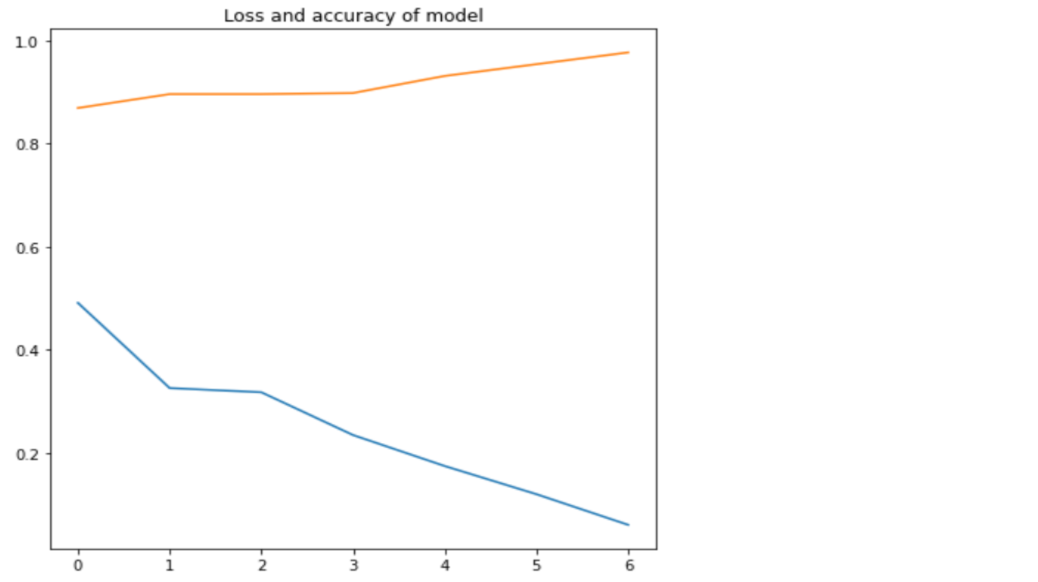
```
In [78]: model = Sequential()
         model.add(Embedding(max_fatures, embed_dim,input_length = X.shape[1]))
         model.add(SpatialDropout1D(0.4))
         model.add(LSTM(lstm_out, dropout=0.2, recurrent_dropout=0.2))
         model.add(Dense(2,activation='softmax'))
         model.compile(loss = 'categorical_crossentropy', optimizer='adam',metrics = ['accuracy'])
         print(model.summary())
         # Here we train the model
         batch_size = 32
         hist = model.fit(X_train, Y_train, epochs = 7, batch_size=batch_size, verbose = 2)
         #Plotting a histogram over the 7 epocs and plotting the accuracy and loss
         history = pd.DataFrame(hist.history)
         plt.figure(figsize=(7,7));
         plt.plot(history["loss"]);
         plt.plot(history["accuracy"]);
         plt.title("Loss and accuracy of model");
         plt.show();
         #Testing the model, and retrieveing score and accuracy:
         score,accuracy = model.evaluate(X_test,Y_test)
         print("score: %.2f" % (score))
         print("accuracy: %.2f" % (accuracy))
         #now we validate for the models accuracy in predicting either a positive, or a negative score:
         validation_size = 1500
```

11. The output of each layer would be an input to a single layer, in the model all layers are stacked in linear fashion. To obtain the required output from the code, 'Sequential()' API has been used which would takes list of layers.

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, 203, 200)          400000
_________________________________________________________________
spatial_dropout1d_2 (Spatial (None, 203, 200)          0
_________________________________________________________________
lstm_2 (LSTM)                (None, 200)               320800
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 402
=================================================================
Total params: 721,202
Trainable params: 721,202
Non-trainable params: 0
```

12. The neural network has been trained using the forward propagation, using loss function to find loss and using optimizer to update weights. In the following output, it can be observed that 7 Epochs have been obtained, Epochs are the number of times the model will iterate in all training examples implemented. Hence, 7 Epochs specify that our model would run for entire dataset 7 times.

```
_________________________________________________________________
None
Epoch 1/7
16/16 - 40s - loss: 0.4908 - accuracy: 0.8693
Epoch 2/7
16/16 - 30s - loss: 0.3254 - accuracy: 0.8963
Epoch 3/7
16/16 - 30s - loss: 0.3172 - accuracy: 0.8963
Epoch 4/7
16/16 - 31s - loss: 0.2341 - accuracy: 0.8983
Epoch 5/7
16/16 - 32s - loss: 0.1740 - accuracy: 0.9315
Epoch 6/7
16/16 - 32s - loss: 0.1193 - accuracy: 0.9544
Epoch 7/7
16/16 - 31s - loss: 0.0598 - accuracy: 0.9772
```




```
8/8 [==============================] - 5s 323ms/step - loss: 0.3633 - accuracy: 0.8782
score: 0.36
accuracy: 0.88
```

The loss and accuracy model has been obtained using the above mentioned code, accuracy of the analysis is 88 % with score of 0.36.

```
1/1 - 0s
1/1 - 0s
1/1 - 0s
positive_acc 96.17224880382776 %
negative_acc 27.586206896551722 %
```

After sentiment analysis, it can be concluded that 96.17 % of the comments on Facebook were **'positive'** and 27.58 % of the comments on Facebook are **'negative'**.

```
In [81]: cmnt = ['your customer service is the absolute worst i now have a mess of books on my kindle']
         #vectorizing the comment
         cmnt = tokenizer.texts_to_sequences(cmnt)
         cmnt = pad_sequences(cmnt, maxlen=203, dtype='int32', value=0)
         print(cmnt)
         sentiment = model.predict(cmnt,batch_size=2,verbose = 2)[0]
         if(np.argmax(sentiment) == 0):
             print("negative")
         elif (np.argmax(sentiment) == 1):
             print("positive")

[[  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0   0   0  51 510 674  12   2 895   1  69  11   6
   13  16  20   3   4]]
1/1 - 0s
positive
```

From above sentiment analysis, it can be concluded that overall, all comments on Facebook are **'positive'** and customers have a very positive sentiment towards the company/brand/product.

REFERENCES –

1. Chatterjee, S., & Krystyanczuk, M. (2017). *Python social media analytics : analyze and visualize data from twitter, youtube, github, and more*. Packt Publishing. https://hult.on.worldcat.org/oclc/999671086

2. Muhammet, S. B., & Fatih, K. (2020*). Sentiment analysis with machine learning methods on social media*, *9*(3), 5–15. https://doi.org/10.14201/ADCAIJ202093515

3. Cambria E., Das D., Bandyopadhyay S., Feraco A. (2017) *Affective Computing and Sentiment Analysis.* In: Cambria E., Das D., Bandyopadhyay S., Feraco A. (eds) *A Practical Guide to Sentiment Analysis. Socio-Affective Computing*, vol 5. Springer, Cham. https://doi.org/10.1007/978-3-319-55394-8_1

APPENDIX –

- **https://github.com/shubhammittal2312/Facebook_Sentiment_Analysis**

- **https://github.com/shubhammittal2312/Facebook_Sentiment_Analysis/blob/main/Indi%20Assign.ipynb**