Neural Network

Stock Prediction & Sentiment Analysis - Part Two

LSTM and Sentiment Analysis

[Udemy]

```
from helpers import *
import numpy as np
import pandas as pd
import re
from importlib import reload
import matplotlib.pyplot as plt
from keras.models import Sequential, load_model
from keras.layers import LSTM, Dense, Embedding, Dropout
from keras.preprocessing.text import Tokenizer
from keras.utils import pad_sequences
from sklearn.model_selection import train_test_split
%matplotlib inline
```

```
2023-01-19 16:19:55.804352: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations:  SSE4.1 SSE4.2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
```

MENU: Top | Importing & Narrowing | Intial Investigation | Preprocessing ||||

Importing and narrowing data

Import

```
tweets = pd.read_csv("Tweets.csv")
```

Shuffle

```
pretty(len(tweets), "Length of dataset before using .sample()");sp()
tweets = tweets.sample(len(tweets)).reset_index(drop=True)
pretty(len(tweets), "Length of dataset after using .sample()")
```

Length of dataset before using .sample()
14640

Length of dataset after using .sample()
14640

Initial view of data

```
pretty(tweets.shape, 'tweets.shape')
head_tail_vert(tweets, 2, "tweets")
```

tweets.shape
(14640, 15)

tweets: head(2)

	tweet_id	airline_sentiment	airline_sentiment_confidence	negativereason	negativereason_confidence	
0	569787479324299265	negative	1.00	Lost Luggage	1.00	Am
1	570158980611350528	negative	1.00	Can't Tell	1.00	Am

tweets: tail(2)

	tweet_id	airline_sentiment	airline_sentiment_confidence	negativereason	negativereason_confidence
14638	569988839919394816	negative	0.65	Can't Tell	0.65
14639	568759575807184896	negative	1.00	Late Flight	1.00

Removing unneeded columns

```
tweets = tweets[['airline_sentiment', 'text']]
```

```
head_tail_vert(tweets, 5, 'Tweets')
```

Tweets: head(5)

	airline_sentiment	text
0	negative	@AmericanAir i've now been in France two days without changing clothes and nothing to keep me warm so please locate my bag and send it to(4)
1	negative	@AmericanAir We all did! Skipper's walk-around too :(
2	negative	@AmericanAir on hold for 3 hours...got an agent, immediately txfd me to change cncdld flight...told it would be 1 min, 80 min Late Flightr, waiting
3	neutral	@VirginAmerica @TTINAC11 I DM you
4	negative	@AmericanAir My reservation is on hold, not me. Wish I was on hold but that's not possible with the phone issues at #americanair

Tweets: tail(5)

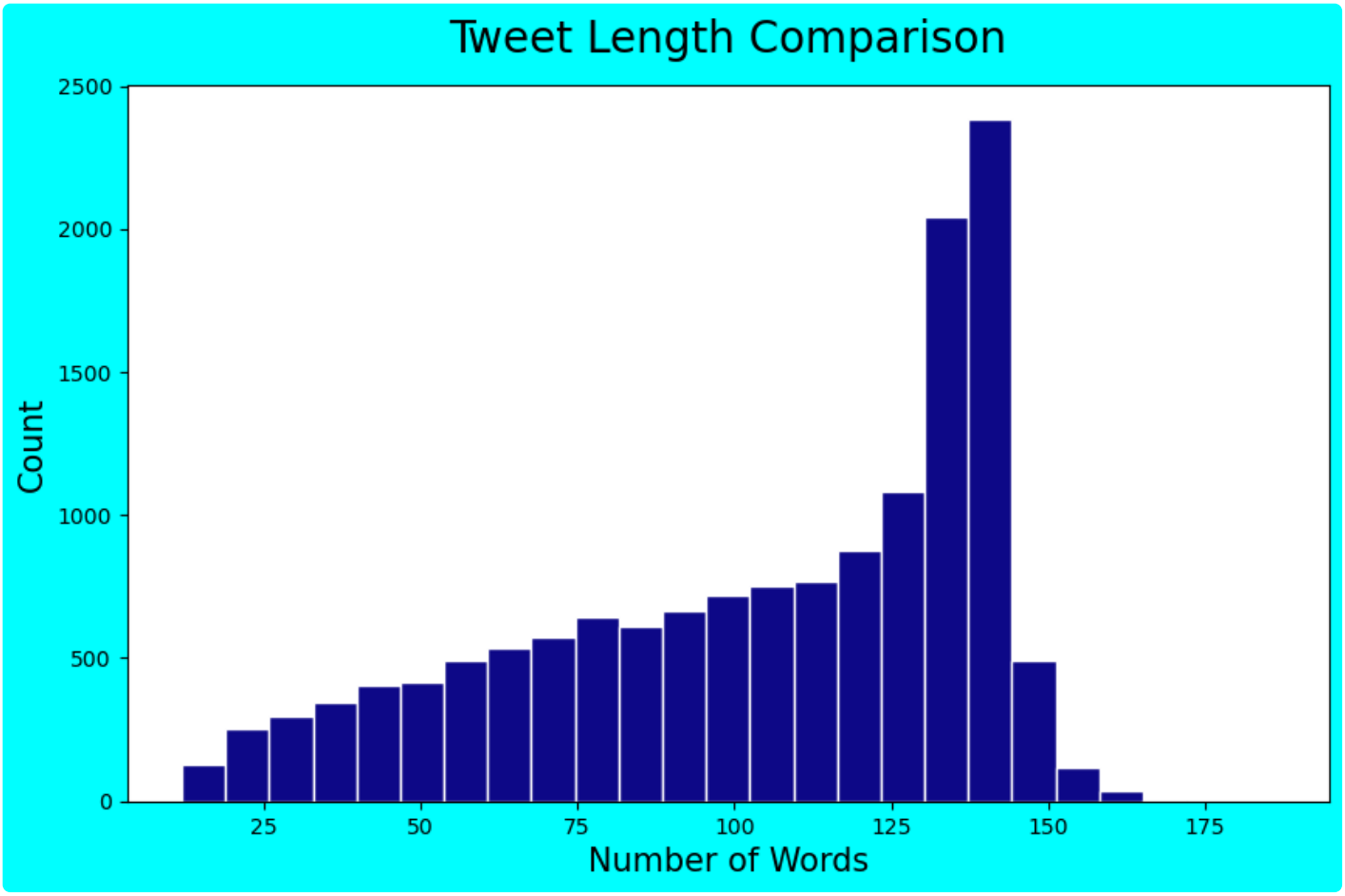
	airline_sentiment	text
14635	positive	@SouthwestAir always when I fly SW. #loyalRRmember
14636	negative	@SouthwestAir I'm running out of money to keep paying for hotel rooms & food in NYC. You don't help people with $ spent
14637	negative	@united why couldn't you have changed the tire of my delayed UA1127 flight when it arrived instead of waiting until boarding?
14638	negative	@united iCloud it is not there yet – PLEASE HELP 917 703 1472
14639	negative	@USAirways today's flight to Philadelphia. It's a little disappointing for the unnecessary delays and it's not even snowing in FL.

MENU: Top | Importing & Narrowing | Intial Investigation | Preprocessing | | | |

Initial Data Analysis

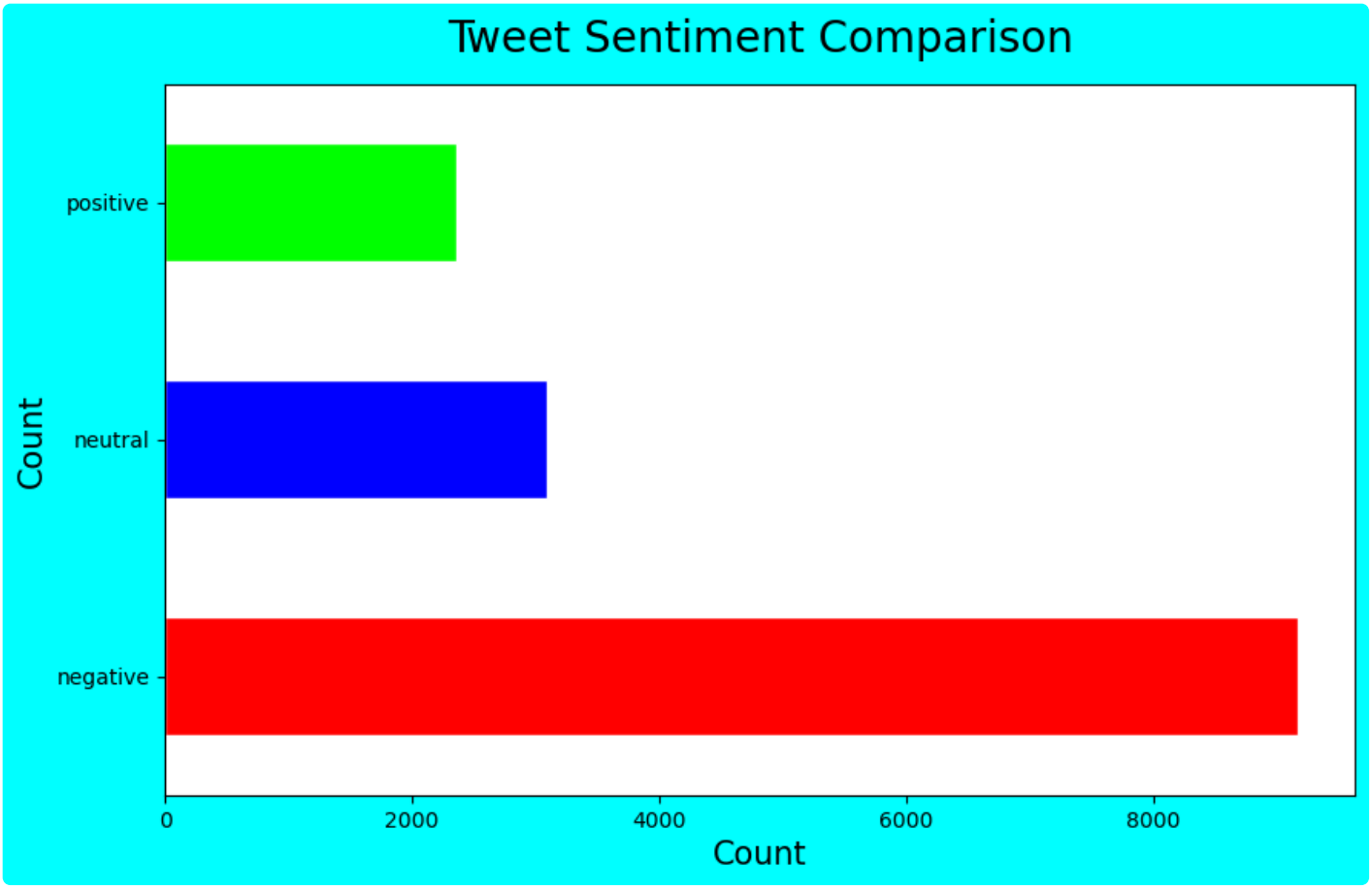
Length of tweets

```
fig = plt.figure(figsize=(10, 6), facecolor="cyan")
tweets.text.str.len().plot.hist(edgecolor="white",
                                cmap="plasma", bins=25);
plt.title("Tweet Length Comparison", fontsize=20, pad=15);
plt.xlabel('Number of Words', fontsize=15); plt.ylabel("Count", fontsize=15);
```



Sentiment Comparison

```
fig = plt.figure(figsize=(10, 6), facecolor="cyan")
tweets.airline_sentiment.value_counts().plot.barh(edgecolor="white",
                                                  color = ['red',
                                                           'blue', 'lime']);
plt.title("Tweet Sentiment Comparison", fontsize=20, pad=15);
plt.xlabel('Count', fontsize=15); plt.ylabel("Count", fontsize=15);
```



MENU: Top | Importing & Narrowing | Intial Investigation | Preprocessing | | | |

Preprocessing

```
see(tweets.text.head(5), "Tweets: Text Column")
```

Tweets: Text Column

	text
0	@AmericanAir i've now been in France two days without changing clothes and nothing to keep me warm so please locate my bag and send it to(4)
1	@AmericanAir We all did! Skipper's walk-around too :(
2	@AmericanAir on hold for 3 hours...got an agent, immediately txfd me to change cncdld flight...told it would be 1 min, 80 min Late Flightr, waiting
3	@VirginAmerica @TTINAC11 I DM you
4	@AmericanAir My reservation is on hold, not me. Wish I was on hold but that's not possible with the phone issues at #americanair

Text cleaning

- making everything lower case
- removing anything that is not alphanumeric using regular expressions
  - passing the characters to remove `[^a-zA-Z0-9\s]`
  - this covers all alpha characters, all numbers, and spaces

- since it is already lowercase, the A-Z is not necessary, but this is often the combination passed for this purpose
- any character that is not one of these will be replaced with nothing, ""

```
tweets.text.apply(lambda x: x.lower())
tweets.text = tweets.text.apply(lambda x: re.sub('[^a-zA-Z0-9\s]', "", x))
```

```
see(tweets.text.head(5), "Tweets: Cleaned Text Columns")
```

Tweets: Cleaned Text Columns

	text
0	AmericanAir ive now been in France two days without changing clothes and nothing to keep me warm so please locate my bag and send it to4
1	AmericanAir We all did Skippers walkaround too
2	AmericanAir on hold for 3 hoursgot an agent immediately txfd me to change cncdld flighttold it would be 1 min 80 min Late Flightr waiting
3	VirginAmerica TTINAC11 I DM you
4	AmericanAir My reservation is on hold not me Wish I was on hold but thats not possible with the phone issues at americanair

Making text inputs uniform length

- `tokenizer` will create our corpus of most frequent words
- `num_words` = the size of the vocabulary, limiting the number of words
- `split` = splitting words on spaces

```
tokenizer = Tokenizer(num_words = 5000, split = " ")
```

Fitting the tokenizer on the text column

- converting text to numpy array with `text.values`

```
tokenizer.fit_on_texts(tweets.text.values)
```

Converting the text / words to numeric values

```
sequences = tokenizer.texts_to_sequences(tweets.text.values)
```

Padding sequences to ensure uniform lengths

```
sequences = pad_sequences(sequences)
```

```
pretty(sequences[0], 'sequences[0]'); sp()
pretty(sequences[1], 'sequences[1]'); sp()
pretty(sequences[2], 'sequences[2]]'); sp()
```

sequences[0]

[0 0 0 0 0 0 0 0 13 113 39 45 15 3665 180 181 325 795 796 10 250 1 274 19 981 44 73 1914 11 88 10 357 20]

sequences[1]

[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 13 38 67 115 167]

sequences[2]]

[0 0 0 0 0 0 0 0 0 0 13 9 62 7 124 40 153 1784 4598 19 1 127 4599 20 76 32 218 213 2718 213 96 238 107]

One-hot encoding the categories of output

```
output = pd.get_dummies(tweets.airline_sentiment).values
```

```
sentiment_categories = pd.concat([tweets.airline_sentiment.iloc[0:6],
                        pd.DataFrame(output[0:6])], axis=1)

see(sentiment_categories, "Examples of one-hot encoded sentiments")
```

Examples of one-hot encoded sentiments

	airline_sentiment	0	1	2
0	negative	1	0	0
1	negative	1	0	0
2	negative	1	0	0
3	neutral	0	1	0
4	negative	1	0	0
5	negative	1	0	0

MENU: Top | Importing & Narrowing | Intial Investigation | Preprocessing | Model | | |

Defining the Model

- `Embedding` :
  - 5,000 is the maximum vocabulary, must be the same as given to tokenizer
  - convert each number (which corresponds to one of the 5,000 words) in the arrays in sequences into a vector of length 256
  - this is how the model groups words with similar meanings or uses, based on these vectors
  - words that are closer together conceptually will be closer together numerically

    - `input_length` is the uniform length that all sequences have been made to fit to by padding
- `Dropout` : to prevent overfitting
- `LSTM`
- `Dense` :
    - has 3 neurons for "postive", "neutral", and "negative" sentiments
    - `softmax` activation, since this is not binary

```
model = Sequential()
model.add(Embedding(5000, 256, input_length=sequences.shape[1]))
model.add(Dropout(0.3))
model.add(LSTM(256, return_sequences = True, dropout=0.3, recurrent_dropout=0.2))
model.add(LSTM(256, dropout = 0.3, recurrent_dropout=0.2))
model.add(Dense(3, activation = 'softmax'))
```

```
2023-01-19 16:20:18.399606: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations:  SSE4.1 SSE4.2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
```

Compiling the model

- `loss = categorical_crossentropy` because this is a classification problem, 3 classes
- `optimizer = 'adam'` for the gradient descent optimization
- `metrics = "accuracy"` is the method by which we will receive feedback about how well the model is performing

```
model.compile(loss = "categorical_crossentropy",
              optimizer = "adam",
              metrics = "accuracy")
```

Summary of the defined model

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 33, 256)           1280000

 dropout (Dropout)           (None, 33, 256)           0

 lstm (LSTM)                 (None, 33, 256)           525312
```

```
 lstm_1 (LSTM)               (None, 256)               525312

 dense (Dense)               (None, 3)                 771

=================================================================
Total params: 2,331,395
Trainable params: 2,331,395
Non-trainable params: 0
_________________________________________________________________
```

Training the Model

Train-Test Split

```
train_in, test_in, train_out, test_out = train_test_split(sequences, output,
                                                          random_state = 123)
```

Fitting the model

```
# model.fit(train_in, train_out,
#         epochs = 10, batch_size = 32)
```

```
Epoch 1/10
344/344 [==============================] - 32s 94ms/step - loss: 0.5528 - accuracy:
0.7768
Epoch 2/10
344/344 [==============================] - 32s 94ms/step - loss: 0.4048 - accuracy:
0.8492
Epoch 3/10
344/344 [==============================] - 31s 91ms/step - loss: 0.3259 - accuracy:
0.8778
Epoch 4/10
344/344 [==============================] - 32s 92ms/step - loss: 0.2581 - accuracy:
0.9032
Epoch 5/10
344/344 [==============================] - 32s 92ms/step - loss: 0.2113 - accuracy:
0.9219
Epoch 6/10
344/344 [==============================] - 32s 93ms/step - loss: 0.1705 - accuracy:
0.9380
Epoch 7/10
```

```
344/344 [==============================] - 32s 92ms/step - loss: 0.1486 - accuracy:
0.9464
Epoch 8/10
344/344 [==============================] - 32s 94ms/step - loss: 0.1302 - accuracy:
0.9533
Epoch 9/10
344/344 [==============================] - 32s 93ms/step - loss: 0.1097 - accuracy:
0.9618
Epoch 10/10
344/344 [==============================] - 32s 93ms/step - loss: 0.1044 - accuracy:
0.9623
```

```
<keras.callbacks.History at 0x7f7b21afa140>
```

Saving the trained model

```
# model.save("sentiment_model.h5")
```

```
model.load("sentiment_model.h5")
```

Getting predictions

```
predictions = model.predict(test_in)
```

```
115/115 [==============================] - 3s 20ms/step
```

MENU: Top | Importing & Narrowing | Intial Investigation | Preprocessing | Defining the Model | Training the Model | Results

---

Results

View prediction results

```
results = pd.concat([tweets[['text', 'airline_sentiment']][0:3660],
                     pd.DataFrame(predictions)[0:3660],
                     pd.DataFrame(test_out)[0:3660]], axis = 1)

results.columns = ['text', 'actual', '%_neg', '%_neu', '%_pos', 'neg', 'neu', 'pos']

results.sample(5)
```

	text	actual	%_neg	%_neu	%_pos	neg	neu	pos
2364	usairways 3rd time cut off after 10 minutes on...	negative	0.01	0.16	0.83	0	0	1
2658	VirginAmerica Your chat support is not working...	negative	1.00	0.00	0.00	0	1	0
1019	united will do Just need to get CVG and my bag...	positive	0.02	0.46	0.53	0	1	0

	text	actual	%_neg	%_neu	%_pos	neg	neu	pos
2094	USAirways on hold with 8004284322 Flight from ...	negative	0.97	0.00	0.02	0	1	0
2385	united How can I file a claim when your agents...	negative	0.54	0.33	0.14	1	0	0

Investigating specific records

```
def view_examples(num, df, preds, cols):
    import random
    for record in range(num):
        random_choice = random.choice(range(0, len(preds)))

        prediction_mask = df.iloc[random_choice][cols]\
        .eq(df.iloc[random_choice][cols].max(), axis=0)

        prediction = str(prediction_mask[prediction_mask==True].index)[8:11]

        actual = df.iloc[random_choice]["actual"]

        pretty(df.iloc[random_choice]['text'],
            f'○ Actual: {actual}    ○ Predicted: {prediction}'); sp()
```

```
view_examples(5, results, predictions, ['neg', 'neu', 'pos'])
```

○ Actual: negative ○ Predicted: neu
united the lounge tells us they have no pillows for my grandma as one of the ladies opens the closet and I see 2 right there unitedlies

○ Actual: positive ○ Predicted: neg
united thanks

○ Actual: neutral ○ Predicted: neg
SouthwestAir first time flyer scheduled a roundtrip set on departure date not sure on returning date policyfees on changing Re Flight

○ Actual: negative ○ Predicted: neg
USAirways 1500 characters are not enough to convey the issue Was directed to emailmailtocustomerrelationsusairwayscom unmonitored BAD

○ Actual: negative ○ Predicted: neg
AmericanAir It is now going to be reported to the police due to the sexual assult sad that you didnt care