

# Sentimental Analysis on Financial Dataset-NLP

```
In [1]: import nltk
#nltk.download_shell()

In [2]: messages = [line.rstrip() for line in open('FinancialData.csv')]
print(len(messages))

4847

In [3]: for message_no, message in enumerate(messages[:10]):
    print(message_no, message)
    print('\n')

0 Response, Features

1 neutral, "According to Gran , the company has no plans to move all production to Russia , although that is where the company is growing ."

2 neutral, "Technopolis plans to develop in stages an area of no less than 100,000 square meters in order to host companies working in computer technologies and telecommunications , the statement said ."

3 negative, "The international electronic industry company Elcoteq has laid off tens of employees from its Tallinn facility ; contrary to earlier layoffs the company contracted the ranks of its office workers , the daily Postimees reported ."

4 positive, "With the new production plant the company would increase its capacity to meet the expected increase in demand and would improve the use of raw materials and therefore increase the production profitability ."

5 positive, "According to the company 's updated strategy for the years 2009-2012 , Basware targets a long-term net sales growth in the range of 20 % -40 % with an operating profit margin of 10 % -20 % of net sales ."

6 positive, "FINANCING OF ASPOCOMP 'S GROWTH Aspocomp is aggressively pursuing its growth strategy by increasingly focusing on technologically more demanding HD I printed circuit boards PCBs ."

7 positive, "For the last quarter of 2010 , Componenta 's net sales doubled to EUR131m from EUR76m for the same period a year earlier , while it moved to a zero pre-tax profit from a pre-tax loss of EUR7m ."

8 positive, "In the third quarter of 2010 , net sales increased by 5.2 % to EUR 205.5 mn , and operating profit by 34.9 % to EUR 23.5 mn ."

9 positive, "Operating profit rose to EUR 13.1 mn from EUR 8.7 mn in the corresponding period in 2007 representing 7.7 % of net sales ."
```

```
In [4]: import pandas as pd
messages = pd.read_csv('FinancialData.csv', names=["label", "message"], encoding='latin-1')
messages.head()
```

	label	message
0	Response	Features
1	neutral	According to Gran , the company has no plans t...
2	neutral	Technopolis plans to develop in stages an area...
3	negative	The international electronic industry company ...
4	positive	With the new production plant the company woul...

```
In [5]: messages.describe()
```

	label	message
count	4847	4847
unique	4	4839
top	neutral	The issuer is solely responsible for the conte...
freq	2879	2

```
In [6]: messages.groupby('label').describe()
```

	count	unique	message	top	freq
label					
Response	1	1	Features	1	
negative	604	604	The international electronic industry company ...	1	
neutral	2879	2873	SSH Communications Security Corporation is hea...	2	
positive	1363	1363	With the new production plant the company woul...	1	

```
In [7]: messages['length'] = messages['message'].apply(len)
messages.head()
```

	label	message	length
0	Response	Features	9
1	neutral	According to Gran , the company has no plans t...	127
2	neutral	Technopolis plans to develop in stages an area...	190
3	negative	The international electronic industry company ...	228
4	positive	With the new production plant the company woul...	206

```
In [8]: import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
messages['length'].plot(bins=50, kind='hist')

<AxesSubplot:ylabel='Frequency'>
```

```
In [9]: messages.length.describe()
```

count	4847.000000
mean	128.107489
std	56.546244
min	9.000000
25%	84.000000
50%	119.000000
75%	163.000000
max	315.000000
Name: length, dtype: float64	

```
In [10]: messages[messages['length'] == 315]['message'].iloc[0]
```

'Supported Nokia phones include : N96 , N95-8GB , N95 , N93-N931 , N92 , N85 , N82 , N81 , N80 , N79 , N78 , N77 , N76 , N75 , N73 , N72 , N71 , E90 , E71 , E70 , E66 , E65 , E62 , E61-E611 , E60 , E51 , E50 , Touch Xpress 5800 , 6220 Classic , 6210 Navigator , 6120 Classic , 6110 Navigator , 5700 , 5500 , 5320XM .'

```
In [11]: messages.head()
```

	label	message	length
0	Response	Features	9
1	neutral	According to Gran , the company has no plans t...	127
2	neutral	Technopolis plans to develop in stages an area...	190
3	negative	The international electronic industry company ...	228
4	positive	With the new production plant the company woul...	206

```
In [12]: import string
from nltk.corpus import stopwords
def text_process(mess):
    """
    Takes in a string of text, then performs the following:
    1. Remove all punctuation
    2. Remove all stopwords
    3. Returns a list of the cleaned text
    """
    # Check characters to see if they are in punctuation
    nopunc = [char for char in mess if char not in string.punctuation]

    # Join the characters again to form the string.
    nopunc = ''.join(nopunc)

    # Now just remove any stopwords
    return [word for word in nopunc.split() if word.lower() not in stopwords.words('english')]
messages['message'].head(5).apply(text_process)
```

	message	length
0	[Features]	9
1	[According, Gran, company, plans, move, produc...	127
2	[Technopolis, plans, develop, stages, area, le...	190
3	[international, electronic, industry, company, ...	228
4	[new, production, plant, company, would, incre...	206

Name: message, dtype: object

```
In [13]: from sklearn.feature_extraction.text import CountVectorizer
bow_transformer = CountVectorizer(analyzer=text_process).fit(messages['message'])

# Print total number of vocab words
print(len(bow_transformer.vocabulary_))
```

12279

```
In [14]: message4 = messages['message'][3]
print(message4)
```

The international electronic industry company Elcoteq has laid off tens of employees from its Tallinn facility ; contrary to earlier layoffs the company contracted the ranks of its office workers , the daily Postimees reported .

```
In [15]: bow4 = bow_transformer.transform([message4])
print(bow4)
print(bow4.shape)
```

(0, 2789)	1
(0, 4854)	1
(0, 5714)	1
(0, 7256)	2
(0, 7412)	1
(0, 7417)	1
(0, 7568)	1
(0, 7887)	1
(0, 7949)	1
(0, 7978)	1
(0, 8218)	1
(0, 8907)	1
(0, 9006)	1
(0, 9148)	1
(0, 9180)	1
(0, 9825)	1
(0, 10493)	1
(0, 10715)	1
(0, 11628)	1
(0, 12187)	1
(1, 12279)	1

```
In [16]: print(bow_transformer.get_feature_names()[6987])
print(bow_transformer.get_feature_names()[12206])
```

capacities  
worth  
C:\Users\User\anaconda3\lib\site-packages\sklearn\utils\deprecation.py:87: FutureWarning: Function get\_feature\_names is deprecated; get\_feature\_names\_out is deprecated in 1.0 and will be removed in 1.2. Please use get\_feature\_names\_out instead.  
warnings.warn(msg, category=FutureWarning)

```
In [17]: messages_bow = bow_transformer.transform(messages['message'])

In [18]: print('Shape of Sparse Matrix: ', messages_bow.shape)
print('Amount of Non-Zero occurrences: ', messages_bow.nnz)
```

Shape of Sparse Matrix: (4847, 12279)  
Amount of Non-Zero occurrences: 60653

```
In [19]: sparsity = (100.0 * messages_bow.nnz / (messages_bow.shape[0] * messages_bow.shape[1]))
print('Sparsity: {}'.format(round(sparsity)))
```

sparsity: 0

```
In [20]: from sklearn.feature_extraction.text import TfidfTransformer

tfidf_transformer = TfidfTransformer().fit(messages_bow)
tfidf4 = tfidf_transformer.transform(bow4)
print(tfidf4)
```

(0, 12187)	0.2252580753229866
(0, 11628)	0.2759349113590728
(0, 10715)	0.18146374644887373
(0, 10493)	0.28927371167701044
(0, 9825)	0.19072152110904086
(0, 9180)	0.20752642208928213
(0, 9148)	0.23632718062223462
(0, 9006)	0.19511437402480764
(0, 8907)	0.17093102579426395
(0, 8218)	0.18911644524845206
(0, 7978)	0.1704834334556429
(0, 7949)	0.21521177352618936
(0, 7887)	0.16670760258901124
(0, 7568)	0.2229883803042969
(0, 7417)	0.28927371167701044
(0, 7412)	0.2591306103780315
(0, 7256)	0.18780013585462332
(0, 5714)	0.23632718062223462
(0, 4854)	0.28927371167701044
(0, 2789)	0.19910684638709505

```
In [21]: messages_tfidf = tfidf_transformer.transform(messages_bow)
print(messages_tfidf.shape)
```

(4847, 12279)

## Traing a model

```
In [23]: from sklearn.naive_bayes import MultinomialNB
spam_detect_model = MultinomialNB().fit(messages_tfidf, messages['label'])

In [24]: print('predicted:', spam_detect_model.predict(tfidf4)[0])
print('expected:', messages.label[1])
```

predicted: neutral  
expected: neutral

## Test the data

```
In [25]: all_predictions = spam_detect_model.predict(messages_tfidf)
print(all_predictions)
```

['neutral' 'neutral' 'neutral' ... 'positive' 'positive' 'neutral']

```
In [26]: from sklearn.metrics import classification_report
print(classification_report(messages['label'], all_predictions))
```

	precision	recall	f1-score	support
Response	0.00	0.00	0.00	1
negative	1.00	0.10	0.18	604
neutral	0.74	0.99	0.85	2879
positive	0.76	0.52	0.62	1363
accuracy			0.75	4847
macro avg	0.62	0.49	0.41	4847
weighted avg	0.78	0.75	0.75	4847

```
In [27]: from sklearn.model_selection import train_test_split

msg_train, msg_test, label_train, label_test = \
train_test_split(messages['message'], messages['label'], test_size=0.2)

print(len(msg_train), len(msg_test), len(msg_train) + len(msg_test))

3877 970 4847

In [28]: from sklearn.pipeline import Pipeline

pipeline = Pipeline([
    ('bow', CountVectorizer(analyzer=text_process)), # strings to token integer counts
    ('tfidf', TfidfTransformer()), # integer counts to weighted TF-IDF scores
    ('classifier', MultinomialNB()), # train on TF-IDF vectors w/ Naive Bayes classifier
])

In [29]: pipeline.fit(msg_train, label_train)
```

```
In [30]: predictions = pipeline.predict(msg_test)
print(classification_report(predictions, label_test))
```

	precision	recall	f1-score	support
negative	0.02	0.67	0.03	3
neutral	0.98	0.69	0.81	811
positive	0.36	0.63	0.46	156
accuracy			0.68	970
macro avg	0.45	0.66	0.43	970
weighted avg	0.88	0.68	0.75	970

```
In [ ]:
```