

FIN3210 Week 5 Assignment Report  
Ma Kexuan 120090651

## Abstract

This report prepared the word cloud, sentiment variable and fog index of a news article, as well as analyzing the Dataset of Tesla.

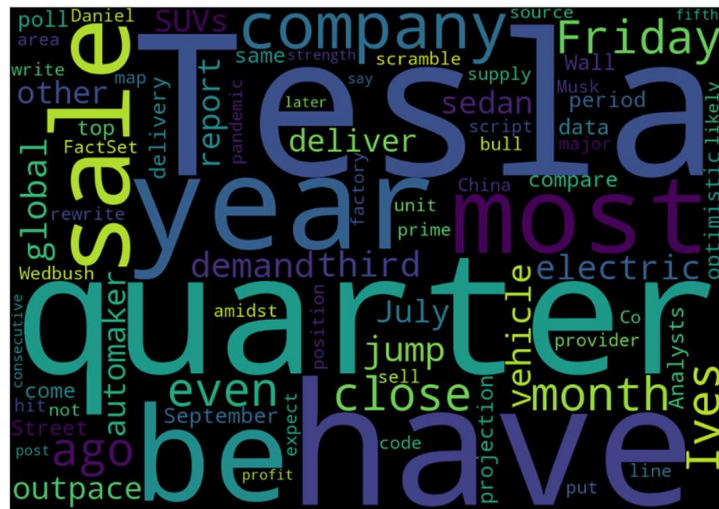
## Data Preprocessing

The preprocessing procedures and some interpretations of the code are described in each code blocks in the appendix, please check.

## Questions

Q1. Present word cloud

The word cloud is shown in the picture below. The news article was originally about the quarterly sales of Tesla. As we can directly see the picture, “Tesla, quarter, sale, company”, those words has the largest font size, which means that they are frequently used in the original article. Through this word cloud, we can see the essential information lucidly.



Q2. Calculate the news sentiment variable using Loughran and McDonald Sentiment Word Lists

By using the positive and negative words included in Loughran and McDonald Sentiment Word Lists, and since I have dealt with the original article into worked clean corpus, then I use the sentiment formula described in tutorial to calculate the overall sentiment of the whole passage. The result is 0.020204, which indicates that the sentiment of the article is quite neutral, and a little bit positive.

Q3. Calculate the Fog index

The Gunning Fog Index is used to measure the difficulty level of reading a passage, the more the Index value, means that the article is more difficult to read. Another interpretation is that the Index value is corresponded to the number of years that is needed to study in order to comprehend the whole article well. The Fog Index of the given article is 12.9282, which means that the article of Tesla is quite hard to comprehend, it approximately needs 13 years of study to better read the whole article.

Q4. Using the data set of Tesla, a) report the summary statistics of sentiment, novelty, and impact; b) present the correlation coefficient among sentiment, novelty, and impact; and c) show the frequency and fraction of top 10 news categories.

a) The summary statistics of sentiment, novelty and impact are provided below, ‘Sentiment’ had an average score of around 53.85 with a standard deviation of 14.82. ‘Novelty’ had an average of 28.88 with a broader spread (standard deviation: 38.54). ‘Impact’ had an average score of 45.28 with a standard deviation of 10.00.

	Sentiment	Novelty	Impact
count	1292.000000	1292.000000	1292.000000
mean	53.845975	28.877709	45.277864
std	14.824257	38.537946	9.996097
min	2.000000	0.000000	13.000000
25%	40.000000	0.000000	39.000000
50%	50.000000	3.000000	45.000000
75%	64.000000	56.000000	52.000000
max	100.000000	100.000000	77.000000

b) The correlation coefficient is provided below, the correlation between ‘Sentiment’ and ‘Novelty’ was 0.1656, suggesting a weak positive relationship. ‘Sentiment’ and ‘Impact’ had a correlation of -0.1310, indicating a weak negative association. The correlation between ‘Novelty’ and ‘Impact’ was -0.0648, suggesting a very weak negative relationship.

	Sentiment	Novelty	Impact
Sentiment	1.000000	0.165557	-0.130972
Novelty	0.165557	1.000000	-0.064844
Impact	-0.130972	-0.064844	1.000000

c) The frequency and fraction of top 10 news categories is provided below, we can see that terms relevant to stocks are frequently shown in the Tesla Dataset, since its stock price is quite volatile and it is genuinely a growing tech company, which makes the result more percipient.

	Category	Frequency	Fractions
0	stock-loss	409	0.316563
1	stock-gain	232	0.179567
2	product-release	111	0.085913
3	business-contract	63	0.048762
4	capital-increase	55	0.042570
5	legal-verdict-favored	42	0.032508
6	price-target-upgrade	40	0.030960
7	fundraising	33	0.025542
8	acquisition-interest-acquirer	33	0.025542
9	product-price-cut	30	0.023220

# FIN3210 Week 5 Assignment

Ma Kexuan

October 20, 2023

```
[1]: import numpy as np
import pandas as pd
import nltk
import matplotlib.pyplot as plt
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from wordcloud import WordCloud
from readability import Readability
```

## 0.0.1 Q1. Present word cloud

```
[2]: # Design a function in order to transfer the tag generated by nltk to
# input in the WordNetLemmatizer
def transfer_to_pos(df):
    if df['Tag'].startswith('J'):
        df['Pos'] = 'a'
    elif df['Tag'].startswith('V'):
        df['Pos'] = 'v'
    elif df['Tag'].startswith('N'):
        df['Pos'] = 'n'
    elif df['Tag'].startswith('R'):
        df['Pos'] = 'r'
    else:
        df['Pos'] = 'x'
    return df
```

```
[3]: article_str = """
Tesla's third-quarter sales jumped 44%\ as
global demand for its electric vehicles outpaced
that of most other automakers. The company reported
Friday that it had delivered 139,000 SUVs and sedans
from July through September, compared with 97,000
deliveries during the same period a year ago. The
sales topped even some of the most optimistic projections
coming from Wall Street. Analysts polled by data provider
FactSet expected the company to sell closer to 137,000.
Tesla has been rewriting the script throughout the year
amidst a pandemic that has closed factories and scrambled
```

```
supply lines. This puts Musk & Co. in prime position to
hit the area code of 500k units for the year which six months
ago was not even on the map for the bulls, Daniel Ives of
Wedbush wrote Friday. China was likely a major source of
strength in the quarter, Ives said. Tesla could post its fifth
consecutive quarter of profits later this month.
"""
```

```
[4]: words_list = word_tokenize(article_str)
words_list[:10]
```

```
[4]: ['Tesla',
      "'s",
      'third-quarter',
      'sales',
      'jumped',
      '44',
      '%',
      '\\',
      'as',
      'global']
```

```
[5]: # Generate the pos_tag of the words list
words_tags = nltk.pos_tag(words_list)
words_tags[:10]
```

```
[5]: [('Tesla', 'NNP'),
      ("'s", 'POS'),
      ('third-quarter', 'JJ'),
      ('sales', 'NNS'),
      ('jumped', 'VBD'),
      ('44', 'CD'),
      ('%', 'NN'),
      ('\\', 'CC'),
      ('as', 'IN'),
      ('global', 'JJ')]
```

```
[6]: words_tags_df = pd.DataFrame(words_tags, columns=['Word_original', 'Tag'])
words_tags_df = words_tags_df.apply(transfer_to_pos, axis=1)
# Only preserve the relevant tags of words
words_tags_df = words_tags_df[words_tags_df['Pos'] != 'x']
words_tags_df = words_tags_df.reset_index(drop=True)
words_tags_df
```

```
[6]:   Word_original  Tag Pos
0         Tesla  NNP   n
1  third-quarter   JJ   a
```

```

2         sales  NNS    n
3         jumped VBD    v
4           %    NN    n
..         ... ..
94    consecutive JJ    a
95         quarter NN    n
96         profits NNS    n
97         later  RB    r
98         month  NN    n

```

[99 rows x 3 columns]

```

[7]: # Lemmatize the words
words_tags_df['words_lemmatized'] = words_tags_df.apply(lambda x:
↳ WordNetLemmatizer().lemmatize(x['Word_original'], pos=x['Pos']), axis=1)
words_tags_df

```

```

[7]:   Word_original  Tag Pos words_lemmatized
0         Tesla  NNP  n         Tesla
1  third-quarter  JJ  a  third-quarter
2         sales  NNS  n         sale
3         jumped VBD  v         jump
4           %    NN  n           %
..         ... ..
94    consecutive JJ  a    consecutive
95         quarter NN  n         quarter
96         profits NNS  n         profit
97         later  RB  r         later
98         month  NN  n         month

```

[99 rows x 4 columns]

```

[8]: # Convert the result dataframe of words into a list
lemmatized_list = words_tags_df['words_lemmatized'].tolist()
lemmatized_str = ' '.join(lemmatized_list)
lemmatized_str

```

```

[8]: 'Tesla third-quarter sale jump % global demand electric vehicle outpace most
other automaker company report Friday have deliver SUVs sedan July September
compare delivery same period year ago sale top even most optimistic projection
come Wall Street Analysts poll data provider FactSet expect company sell close
Tesla have be rewrite script year amidst pandemic have close factory scramble
supply line put Musk Co. prime position hit area code unit year month ago be not
even map bull Daniel Ives Wedbush write Friday China be likely major source
strength quarter Ives say Tesla post fifth consecutive quarter profit later
month'

```



```

349      WIN
350     WINNER
351    WINNERS
352   WINNING
353   WORTHY

```

```
[354 rows x 1 columns]
```

```
[11]: neg_words_list = pd.read_excel('LoughranMcDonald_SentimentWordLists_2018.xlsx',
    ↪sheet_name='Negative', header=None)
neg_words_list
```

```

[11]:      0
0      ABANDON
1     ABANDONED
2    ABANDONING
3   ABANDONMENT
4  ABANDONMENTS
...
2350  WRONGDOING
2351  WRONGDOINGS
2352   WRONGFUL
2353  WRONGFULLY
2354   WRONGLY

```

```
[2355 rows x 1 columns]
```

```
[12]: # If positive, give a 1; If negative give -1; Else 0
def judge_sentiment(df):
    if df['words_lemmatized'].upper() in pos_words_list[0].tolist():
        df['sentiment'] = 1
    elif df['words_lemmatized'].upper() in neg_words_list[0].tolist():
        df['sentiment'] = -1
    else:
        df['sentiment'] = 0
    return df
```

```
[13]: words_tags_df = words_tags_df.apply(judge_sentiment, axis=1)
# Sum the positive and negative weights together
num = words_tags_df['sentiment'].sum()
denom = len(words_tags_df)
new_sentiment = num / denom
new_sentiment
```

```
[13]: 0.020202020202020204
```

### 0.0.3 Q3. Calculate the Fog index

```
[14]: # Calculate the Gunning Fog Index using Readability
r = Readability(article_str)
fog_index = r.gunning_fog().score
fog_index
```

```
[14]: 12.928205128205128
```

### 0.0.4 Q4. Using the data set of Tesla, a) report the summary statistics of sentiment, novelty, and impact; b) present the correlation coefficient among sentiment, novelty, and impact; and c) show the frequency and fraction of top 10 news categories.

```
[15]: # Provide the statistics required in the Homework requirement
tesla_data = pd.read_excel('FIN3210 Week 5 Tesla.xlsx')
summary_stat_data = tesla_data[['Sentiment', 'Novelty', 'Impact']]
summary_stat = summary_stat_data.describe()
summary_stat
```

```
[15]:
```

	Sentiment	Novelty	Impact
count	1292.000000	1292.000000	1292.000000
mean	53.845975	28.877709	45.277864
std	14.824257	38.537946	9.996097
min	2.000000	0.000000	13.000000
25%	40.000000	0.000000	39.000000
50%	50.000000	3.000000	45.000000
75%	64.000000	56.000000	52.000000
max	100.000000	100.000000	77.000000

```
[16]: summary_stat_data.corr()
```

```
[16]:
```

	Sentiment	Novelty	Impact
Sentiment	1.000000	0.165557	-0.130972
Novelty	0.165557	1.000000	-0.064844
Impact	-0.130972	-0.064844	1.000000

```
[17]: news_categories = tesla_data['Category'].value_counts().reset_index()
news_categories.columns = ['Category', 'Frequency']
news_categories['Fractions'] = news_categories['Frequency'] /
↳ news_categories['Frequency'].sum()
news_categories[:10]
```

```
[17]:
```

	Category	Frequency	Fractions
0	stock-loss	409	0.316563
1	stock-gain	232	0.179567
2	product-release	111	0.085913
3	business-contract	63	0.048762



4	capital-increase	55	0.042570
5	legal-verdict-favored	42	0.032508
6	price-target-upgrade	40	0.030960
7	fundraising	33	0.025542
8	acquisition-interest-acquirer	33	0.025542
9	product-price-cut	30	0.023220