blockchain_feelings

July 14, 2020

1 The Feelings of the Blockchain

When you read a news article, normally the headline is the hook to continue reading. However, a negative title could lead you to skip reading an article if you don't want to be in a bad mood. But is this accurate?

On this activity you are tasked to corroborate if a news title with a negative sentiment leads or not to a negative content. You will use VADER sentiment to accomplish this work using the news articles that you previously download on *The Voice of the Blockchain* activity.

[1]:
```
# Initial imports
from path import Path
import pandas as pd
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer

get_ipython().run_line_magic("matplotlib", "inline")
```

1.1 Instructions

Just for convenience download the `vader_lexicon` in order to initialize the VADER sentiment analyzer

[2]:
```
# Download/Update the VADER Lexicon
nltk.download("vader_lexicon")

# Initialize the VADER sentiment analyzer
analyzer = SentimentIntensityAnalyzer()
```

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     /Users/Andrew/nltk_data…
[nltk_data]   Package vader_lexicon is already up-to-date!
```

1.1.1 Load the News Articles from the CSV File as a DataFrame

Pick the CSV file you created on *The Voice of the Crisis* activity and load it as a DataFrame, remember to specify the `encoding='utf-8-sig'` parameter.

[3]:
```
# Load news from CSV file
file_path = Path("Data/blockchain_news_en_fr.csv")
news_df = pd.read_csv(file_path, encoding="utf-8-sig")
news_df.head()
```

[3]:

	title	description	text	date	language
0	Hedge Fund Manager Horizon Kinetics Expands Th…	Core Scientific announced yesterday that the h…	Bitcoin cryptocurrency representation is seen …	2020-07-02	en
1	Global Blockchain Technology Industry - GlobeN…	Global Blockchain Technology Market to Reach U…	New York, July 10, 2020 (GLOBE NEWSWIRE) -- Re…	2020-07-10	en
2	Blockchain Bites: Digital Dollars, Ethereum's …	Thought leaders are decending on Washing to ta…	Australia's stock exchange may delay its block…	2020-06-30	en
3	Deloitte's 2020 Global Blockchain Survey: Near…	Now in its third year, Deloitte's "Global Bloc…	Leaders are increasingly investing in blockcha…	2020-06-16	en
4	Binance Gives Back: How The World's Largest Cr…	Binance Charity launches a "fully transparent"…	The pandemic has changed the world as we know …	2020-06-25	en

The VADER sentiment module is only trained to score sentiment on English language, so create a new DataFrame only with news in English. You will learn how to score sentiment in multiple languages later.

[4]:
```
# Fetch only English news
news_en_df = news_df[news_df["language"] == "en"]
news_en_df.head()
```

[4]:

	title	description
0	Hedge Fund Manager Horizon Kinetics Expands Th…	Core Scientific announced yesterday that the h…
1	Global Blockchain Technology Industry - GlobeN…	Global Blockchain Technology Market to Reach U…
2	Blockchain Bites: Digital Dollars, Ethereum's …	Thought leaders are decending on Washing to ta…
3	Deloitte's 2020 Global Blockchain Survey: Near…	Now in its third year, Deloitte's "Global Bloc…
4	Binance Gives Back: How The World's Largest Cr…	Binance Charity launches a "fully transparent"…

	text	date	language
0	Bitcoin cryptocurrency representation is seen …	2020-07-02	en
1	New York, July 10, 2020 (GLOBE NEWSWIRE) -- Re…	2020-07-10	en
2	Australia's stock exchange may delay its block…	2020-06-30	en
3	Leaders are increasingly investing in blockcha…	2020-06-16	en
4	The pandemic has changed the world as we know …	2020-06-25	en

1.1.2 Calculating VADER Sentiment Score for News Titles and Text

As you know the `compound` score could be used to get a normalized score for a sentiment, in this section you have to create a function called `get_sentiment(score)` that will return a normalized value of sentiment for the `score` parameter based on the rules you learn. This function should return `1` for positive sentiment, `-1` for negative sentiment, and `0` for neutral sentiment.

[5]:
```
# Sentiment calculation based on compound score
def get_sentiment(score):
    """
    Calculates the sentiment based on the compound score.
    """
    result = 0  # Neutral by default
    if score >= 0.05:  # Positive
        result = 1
    elif score <= -0.05:  # Negative
        result = -1

    return result
```

Use the the VADER sentiment module from `NLTK` to score the sentiment of every news article title and text in english; you should append ten new columns to the English news DataFrame to store the results as follows.

- Title's compound score
- Title's positive score
- Title's neutral score
- Title's negative score
- Title's normalized score (using the `get_sentiment()` function)
- Text's compound score
- Text's positive score
- Text's neutral score
- Text's negative score
- Text's normalized score (using the `get_sentiment()` function)

[6]:
```
# Sentiment scores dictionaries
title_sent = {
    "title_compound": [],
    "title_pos": [],
    "title_neu": [],
```

```
    "title_neg": [],
    "title_sent": [],
}
text_sent = {
    "text_compound": [],
    "text_pos": [],
    "text_neu": [],
    "text_neg": [],
    "text_sent": [],
}

# Get sentiment for the text and the title
for index, row in news_en_df.iterrows():
    try:
        # Sentiment scoring with VADER
        title_sentiment = analyzer.polarity_scores(row["title"])
        title_sent["title_compound"].append(title_sentiment["compound"])
        title_sent["title_pos"].append(title_sentiment["pos"])
        title_sent["title_neu"].append(title_sentiment["neu"])
        title_sent["title_neg"].append(title_sentiment["neg"])
        title_sent["title_sent"].
 ↪append(get_sentiment(title_sentiment["compound"]))

        text_sentiment = analyzer.polarity_scores(row["text"])
        text_sent["text_compound"].append(text_sentiment["compound"])
        text_sent["text_pos"].append(text_sentiment["pos"])
        text_sent["text_neu"].append(text_sentiment["neu"])
        text_sent["text_neg"].append(text_sentiment["neg"])
        text_sent["text_sent"].append(get_sentiment(text_sentiment["compound"]))
    except AttributeError:
        pass

# Attaching sentiment columns to the News DataFrame
title_sentiment_df = pd.DataFrame(title_sent)
text_sentiment_df = pd.DataFrame(text_sent)
news_en_df = news_en_df.join(title_sentiment_df).join(text_sentiment_df)

news_en_df.head()
```

[6]:

```
                                               title  \
0  Hedge Fund Manager Horizon Kinetics Expands Th…
1  Global Blockchain Technology Industry - GlobeN…
2  Blockchain Bites: Digital Dollars, Ethereum's …
3  Deloitte's 2020 Global Blockchain Survey: Near…
4  Binance Gives Back: How The World's Largest Cr…

                                         description  \
```

```
0  Core Scientific announced yesterday that the h…
1  Global Blockchain Technology Market to Reach U…
2  Thought leaders are decending on Washing to ta…
3  Now in its third year, Deloitte's "Global Bloc…
4  Binance Charity launches a "fully transparent"…
```

	text	date	language	\
0	Bitcoin cryptocurrency representation is seen …	2020-07-02	en	
1	New York, July 10, 2020 (GLOBE NEWSWIRE) -- Re…	2020-07-10	en	
2	Australia's stock exchange may delay its block…	2020-06-30	en	
3	Leaders are increasingly investing in blockcha…	2020-06-16	en	
4	The pandemic has changed the world as we know …	2020-06-25	en	

	title_compound	title_pos	title_neu	title_neg	title_sent	text_compound	\
0	0.4404	0.231	0.769	0.0	1	0.0000	
1	0.0000	0.000	1.000	0.0	0	0.0000	
2	0.0000	0.000	1.000	0.0	0	-0.3182	
3	0.4215	0.128	0.872	0.0	1	0.3612	
4	0.4767	0.181	0.819	0.0	1	0.3400	

	text_pos	text_neu	text_neg	text_sent
0	0.000	1.000	0.000	0
1	0.000	1.000	0.000	0
2	0.000	0.931	0.069	-1
3	0.099	0.901	0.000	1
4	0.057	0.943	0.000	1

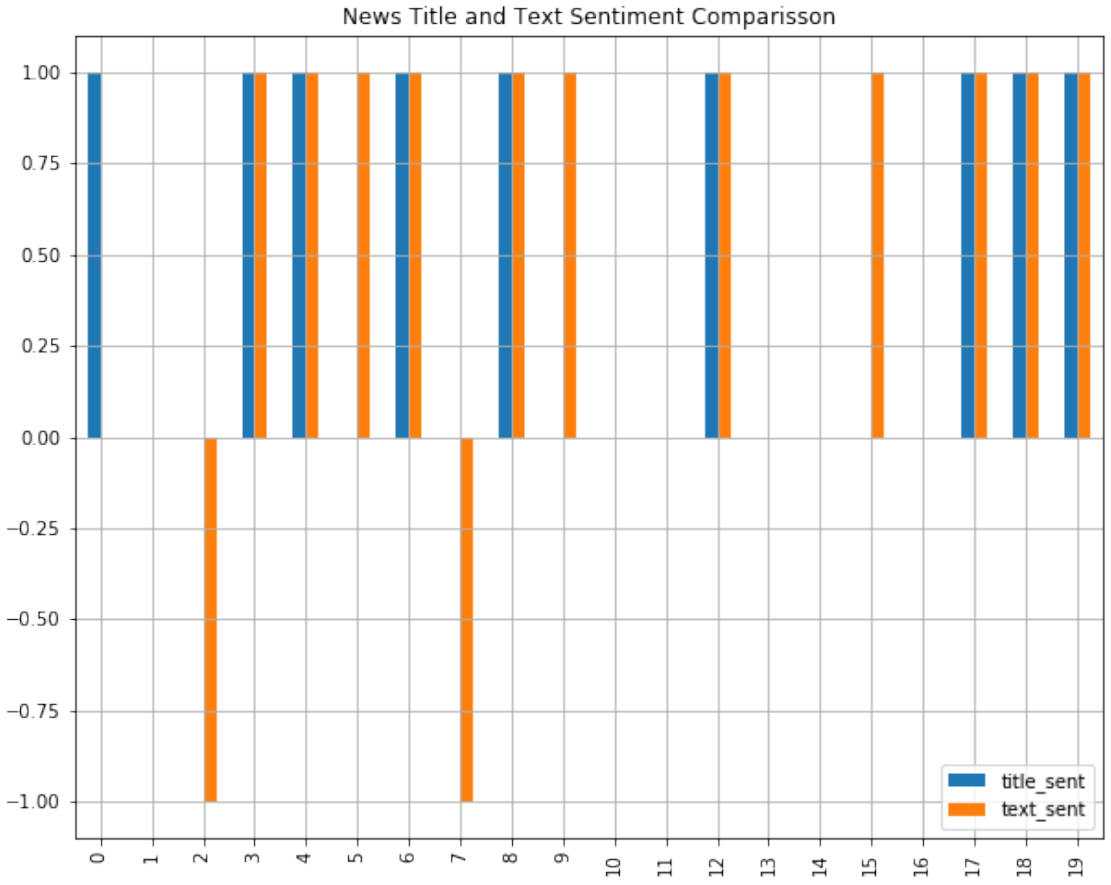
1.1.3 Analyzing Sentiments Results

How the sentiment of the title and the text differs on news articles?

To answer this question, on this section you will create a bar chart contrasting the normalized sentiment for the title and the text of each news article. Use the build-in `plot()` method of the Pandas DataFrame to create a bar chart like the one bellow. Be aware that you chart might differ from this one due to is made from a different news DataFrame.

[7]:
```
# Plot bar chart
news_en_df.plot(
    y=["title_sent", "text_sent"],
    kind="bar",
    title="News Title and Text Sentiment Comparisson",
    figsize=(10, 8),
    grid=True,
)
```

[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a1c5c0a50>
```



Finally get the descriptive statistics from the English news DataFrame and discuss the analysis results with your partners.

[8]:
```
# Describe dataframe
news_en_df.describe()
```

[8]:

	title_compound	title_pos	title_neu	title_neg	title_sent	\
count	20.000000	20.00000	20.00000	20.0	20.000000	
mean	0.214150	0.08305	0.91695	0.0	0.450000	
std	0.269250	0.10424	0.10424	0.0	0.510418	
min	0.000000	0.00000	0.72100	0.0	0.000000	
25%	0.000000	0.00000	0.82950	0.0	0.000000	
50%	0.000000	0.00000	1.00000	0.0	0.000000	
75%	0.426225	0.17050	1.00000	0.0	1.000000	
max	0.735100	0.27900	1.00000	0.0	1.000000	

	text_compound	text_pos	text_neu	text_neg	text_sent
count	20.000000	20.000000	20.000000	20.000000	20.000000
mean	0.222300	0.067900	0.916750	0.015350	0.450000
std	0.314711	0.072375	0.080301	0.031725	0.686333

min	-0.318200	0.000000	0.768000	0.000000	-1.000000
25%	0.000000	0.000000	0.883500	0.000000	0.000000
50%	0.295700	0.062500	0.931500	0.000000	1.000000
75%	0.371375	0.107750	1.000000	0.000000	1.000000
max	0.890800	0.232000	1.000000	0.085000	1.000000