Sentiment Analysis on publicly traded companies

J016 - Avneesh Dubey
J054 - Aayush Talekar

Table of Contents

What is Sentiment Analysis?

- Sentiment analysis is the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information.
- Sentiment analysis (in Finance) involves quantifying and exploiting ‘sentiment’/’emotions’ for investment.
- ‘Sentiment’ can include (but is not limited to) positivity, negativity, uncertainty, narcissism, anxiety, panic, etc.
- There are broadly 2 ways of estimating ‘sentiment’:
  1. Lexicon / Dictionary based approach,
  2. “Machine learning” approach

Lexicon / Dictionary Approach

- In the lexicon / dictionary based approach, we start with a ‘prior’ on what constitutes words relating to ‘sentiment’ (e.g.,
- ‘positivity’, ‘negativity’, etc).
- We use a “sentiment language” (ψ).
- We then estimate ‘sentiment’ (ϕ) as a function of the (cleaned) words ($\mathscr{W}*$) in a given document ($\mathscr{d}$) which belong to a sentiment language (ψ).
- As far as estimating sentiment in Finance goes, the literature largely relies on lexicon / dictionary based approach.

Machine Learning Approach

- In the “Machine learning” approach, we start with a subsample of the ‘corpus’ $\mathcal{C}$ which displays ‘sentiment’ (ϕ).
- E.g., A range of movie reviews ‘labelled’ as either ‘positive’ or ‘negative’.
- We then apply ‘machine learning’ (e.g. classification) algorithms to categorise / classify other sample text on its level of ‘sentiment’ (ϕ).
- Applying this approach in Finance is problematic though, because we don’t have data with sentiment ‘labels’.
- E.g. there’s no database of companies that are
- already classified by ‘sentiment’

The SEC

- The **U.S. Securities and Exchange Commission** (**SEC**) is a large independent agency of the United States federal government that was created following the stock market crash in the 1930s to protect investors and the national banking system.
- The primary purpose of the SEC is to enforce the law against market manipulation.
- The SEC has a three-part mission: to protect investors; maintain fair, orderly, and efficient markets; and facilitate capital formation.[5]
- To achieve its mandate, the SEC enforces the statutory requirement that public companies and other regulated companies submit quarterly and annual reports, as well as other periodic reports.
- In addition to annual financial reports, company executives must provide a narrative account, called the "management discussion and analysis" (MD&A), that outlines the previous year of operations and explains how the company fared in that time period.
- MD&A will usually also touch on the upcoming year, outlining future goals and approaches to new projects.
- In an attempt to level the playing field for all investors, the SEC maintains an online database called EDGAR (the Electronic Data Gathering, Analysis, and Retrieval system) online from which investors can access this and other information filed with the agency.

The SEC's EDGAR Database

- Quarterly and semiannual reports from public companies are crucial for investors to make sound decisions when investing in the capital markets.
- Unlike banking, investment in the capital markets is not guaranteed by the federal government.
- The potential for big gains needs to be weighed against that of sizable losses.
- Mandatory disclosure of financial and other information about the issuer and the security itself gives private individuals as well as large institutions the same basic facts about the public companies they invest in, thereby increasing public scrutiny while reducing insider trading and fraud.
- The SEC makes reports available to the public through the EDGAR system. The SEC also offers publications on investment-related topics for public education.
- The same online system also takes tips and complaints from investors to help the SEC track down violators of the securities laws.
- The SEC adheres to a strict policy of never commenting on the existence or status of an ongoing investigation.

SEC 10-K Filings:

- A 10-K is a comprehensive report filed annually by a publicly-traded company about its financial performance and is required by the U.S. Securities and Exchange Commission (SEC).
- The report contains much more detail than a company's annual report, which is sent to its shareholders before an annual meeting to elect company directors.
- Some of the information a company is required to document in the 10-K includes its history, organizational structure, financial statements, earnings per share, subsidiaries, executive compensation, and any other relevant data.

More on 10-K filings:

Often, the most essential components of the annual 10-K filing include:

- Item 1: Business (a description of the company's operation)
- Item 1A: Risk Factors
- Item 3: Legal Proceedings
- Item 6: Selected Financial Data
- Item 7: Management's Discussion and Analysis of the Financial Condition

The Components we focus on:

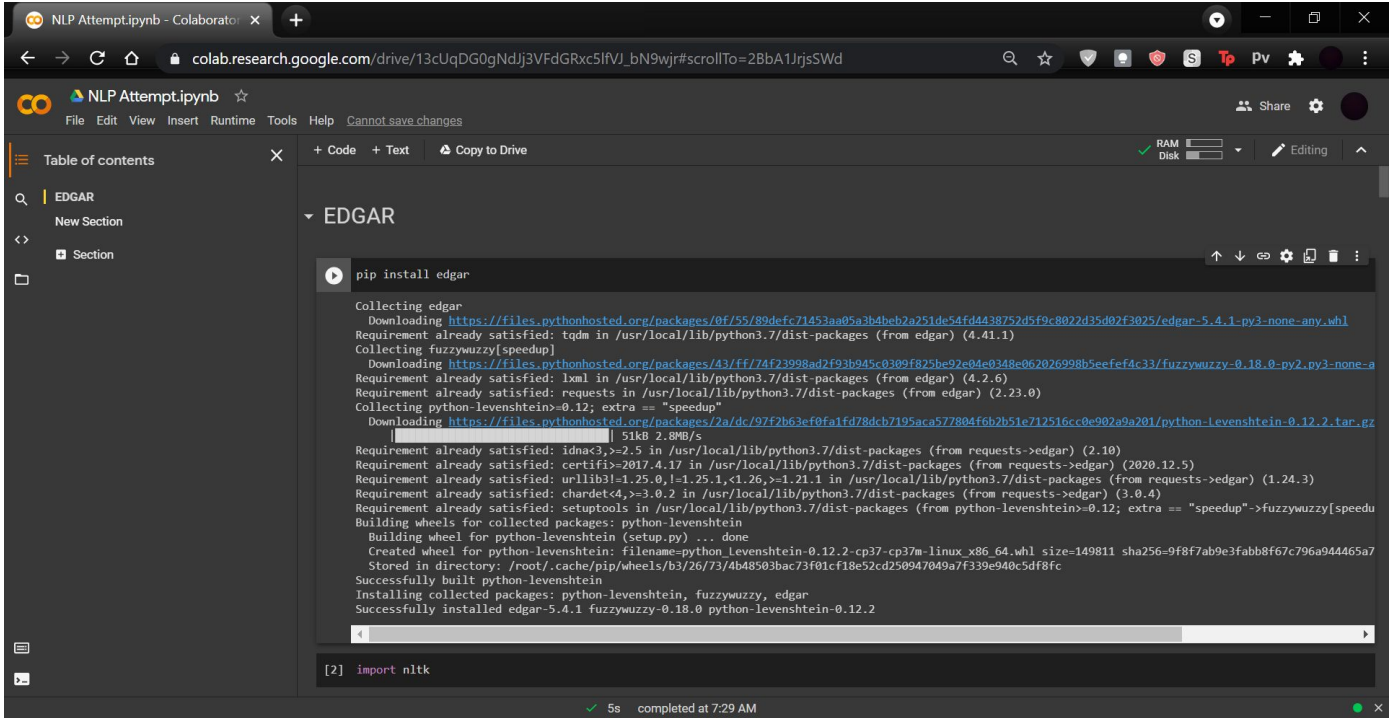
1

Item 1A - Risk factors. These outline any and all risks the company faces or may face in the future. The risks are typically listed in order of importance.

2

Item 7 - Management's discussion and analysis of financial condition and results of operations. Also known as MD&A, this gives the company an opportunity to explain its business results from the previous fiscal year. This section is where the company can tell its story in its own words.

Initial Project:



NLP Attempt.ipynb - Colaborator

colab.research.google.com/drive/1Kq8_dXkCk_AtmxwD-w-NtG_CiBbaVVH3#scrollTo=kDr6pU0Ksjbh

NLP Attempt.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

Table of contents

EDGAR

New Section

Section

+ Code + Text

Connect Editing

```
import re
#Using regular expressions we can find the places where Item 1A. is referenced, the risk section
#Notice the backlash has to be used for the period to escape it
#There are two places we find it, the table of contents and the header
print(re.findall(re.compile('Item 1A\.'),sample_text))
```

```
['Item 1A.']
```

```
#A better way is to search and find the elements. This will give location as well as matched string
for match in re.finditer('Item 1A\.', sample_text):
    print(match)
```

```
<re.Match object; span=(26459, 26467), match='Item 1A.'>
```

```
#We can grab the span as so
print(match.span())
```

```
(26459, 26467)
```

```
#We can also generalize the regular expression to all items
matches = list(re.finditer(re.compile('Item [0-9][A-Z]*\.'), sample_text))
print(matches)
```

```
[<re.Match object; span=(26459, 26467), match='Item 1A.'>, <re.Match object; span=(26480, 26488), match='Item 1B.'>, <re.Match object; span=(26515, 26522
```

```
#The first index of the mathces will be the actual string
print(matches[1][0])
```

```
Item 1B.
```

Where we faced issues.

NLP Attempt.ipynb - Colaborator

colab.research.google.com/drive/1Kq8_dXkCk_AtmxwD-w-NtG_CiBbaVVH3#scrollTo=kDr6pU0Ksjbh

NLP Attempt.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

Table of contents

EDGAR

New Section

Section

+ Code + Text

Connect Editing

```
[ ] get_threat_level_of_word(company_name="Oracle Corp", company_id="0001341439", word='loss')

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-43-80b4aaad5143> in <module>()
----> 1 get_threat_level_of_word(company_name="Oracle Corp", company_id="0001341439", word='loss')

                              2 frames
<ipython-input-19-75957012d47b> in pull_risk_section(text)
      4     text = re.sub('\xa0', ' ', text)
      5     matches = list(re.finditer(re.compile('Item [0-9][A-Z]*\.'), text))
----> 6     start = max([i for i in range(len(matches)) if matches[i][0] == 'Item 1B.'])
      7     end = start+1
      8     start = matches[start].span()[1]

ValueError: max() arg is an empty sequence

SEARCH STACK OVERFLOW
```

```
[ ] negative = pd.read_csv('https://raw.githubusercontent.com/AayushTalekar/NLP/main/Negative.csv', header=None)
    positive = pd.read_csv('https://raw.githubusercontent.com/AayushTalekar/NLP/main/Positive.csv', header=None)
    constraining = pd.read_csv('https://raw.githubusercontent.com/AayushTalekar/NLP/main/Constraining.csv', header=None)
    uncertainity = pd.read_csv('https://raw.githubusercontent.com/AayushTalekar/NLP/main/Uncertainity.csv', header=None)
    weakmodal = pd.read_csv('https://raw.githubusercontent.com/AayushTalekar/NLP/main/Weak%20Modal.csv', header=None)
    strongmodal = pd.read_csv('https://raw.githubusercontent.com/AayushTalekar/NLP/main/Strong%20Modal.csv', header=None)
    litigious = pd.read_csv('https://raw.githubusercontent.com/AayushTalekar/NLP/main/Litigious.csv', header=None)
```

```
[ ] negative.columns = ['Word']
    positive.columns = ['Word']
    constraining.columns = ['Word']
    uncertainity.columns = ['Word']
    weakmodal.columns = ['Word']
```

Link: https://colab.research.google.com/drive/1Kq8_dXkCk_AtmxwD-w-NtG_CiBbaVVH3?usp=sharing

Constructing a Document-Term Matrix via Sklearn, NLTK

- A Document-Term Matrix is used as a starting point for a number of NLP tasks. This short write up shows how to use Sklearn and NLTK python libraries to construct frequency and binary versions.

Constructing a Document-Term Matrix via Sklearn, NLTK

- Importing Libraries:

```
import re

import pandas as pdfrom nltk import word_tokenize

from nltk.corpus import stopwords

from nltk.tokenize import word_tokenize

from nltk.stem.porter import PorterStemmerfrom sklearn.feature_extraction.text import
CountVectorizer
```

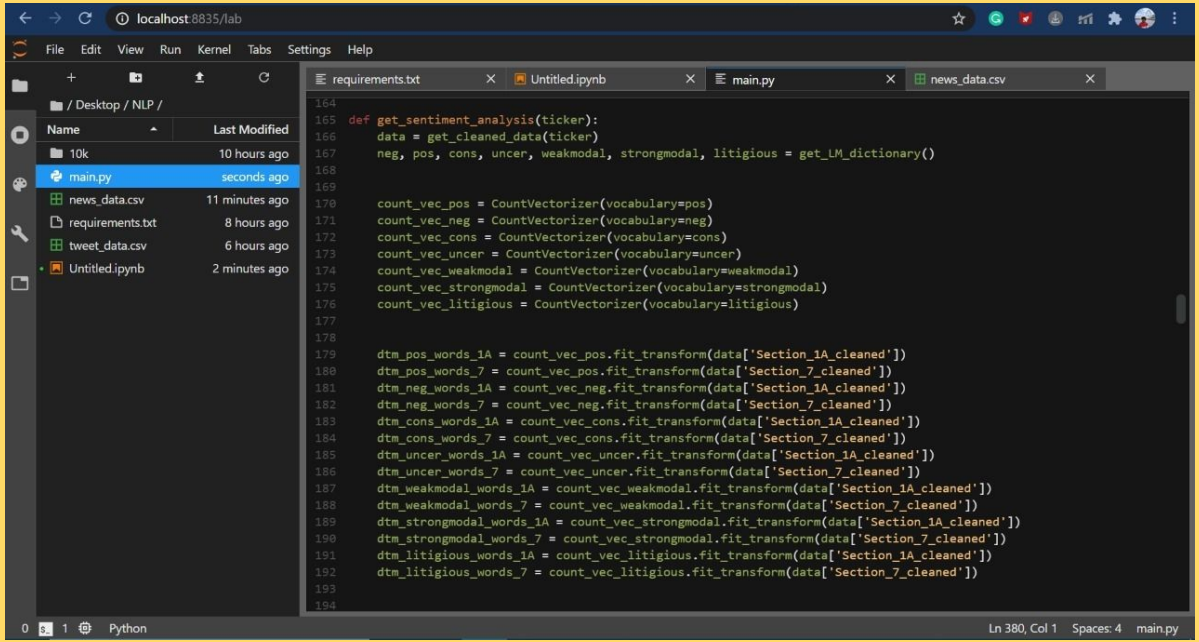
Constructing a Document-Term Matrix via Sklearn, NLTK

- Importing Libraries:

```
import re

import pandas as pdfrom nltk import word_tokenize

from nltk.corpus import stopwords

from nltk.tokenize import word_tokenize

from nltk.stem.porter import PorterStemmerfrom sklearn.feature_extraction.text import
CountVectorizer
```

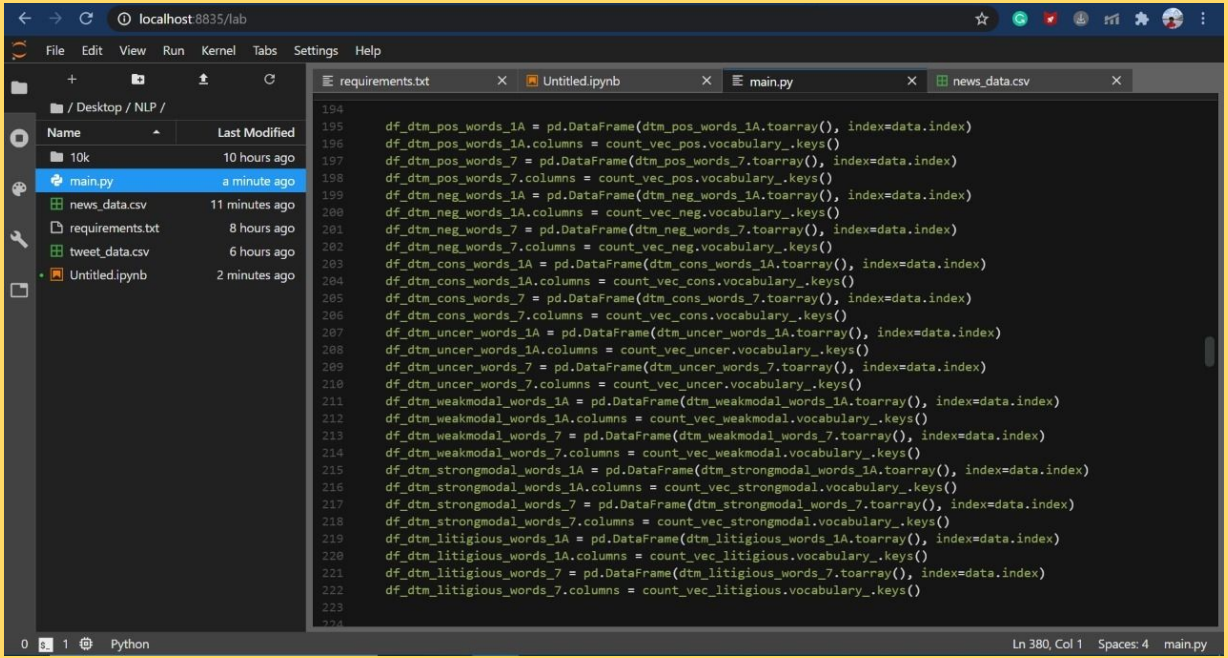
Constructing a Document-Term Matrix via Sklearn, NLTK

- Setting up Tokenizer:

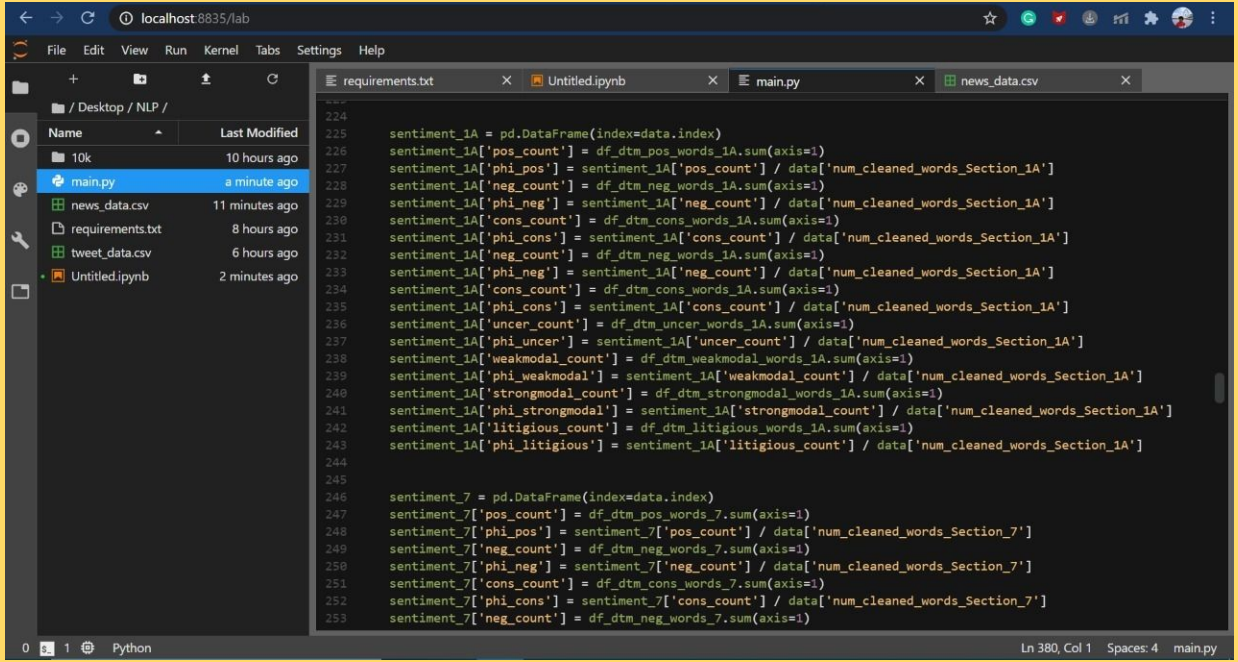


Constructing a Document-Term Matrix via Sklearn, NLTK

- Document-Term Frequency Matrix:



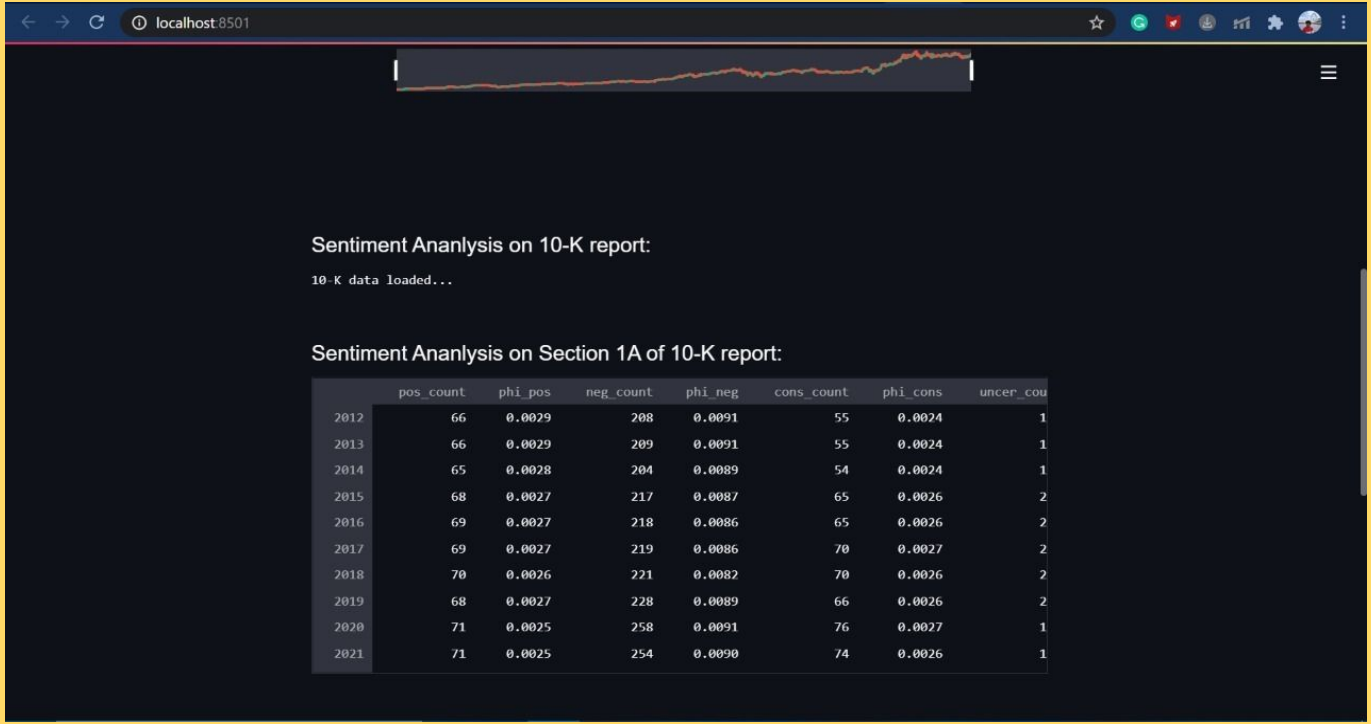
Constructing a Document-Term Matrix via Sklearn, NLTK



Sentiment Analysis using finBERT

- BERT was the language model that made the whole concept of pre-training and fine-tuning flow very popular.
- BERT brought two core innovations to language modelling: (1) It borrowed the **transformer** (T of BERT) architecture from machine translation, which does a better job of modelling long-term dependencies than RNN-based ones (excellent overview here). (2) It introduced the Masked Language Modelling (MLM) task, where a random 15% of all tokens are masked and the model predicts them, enabling true **bi-directionality** (B of BERT).
- BERT was perfect for our task of financial sentiment analysis. Even with a very small dataset, it was now possible to take advantage of state-of-the-art NLP models. But since our domain — finance is very different from the general purpose corpus BERT was trained on, we wanted to add one more step before going for sentiment analysis. Pre-trained BERT knew how to talk, but now it was time to teach it how to *talk like a trader*. We took the pre-trained BERT and then further trained it on a purely financial corpus.

Final Project:

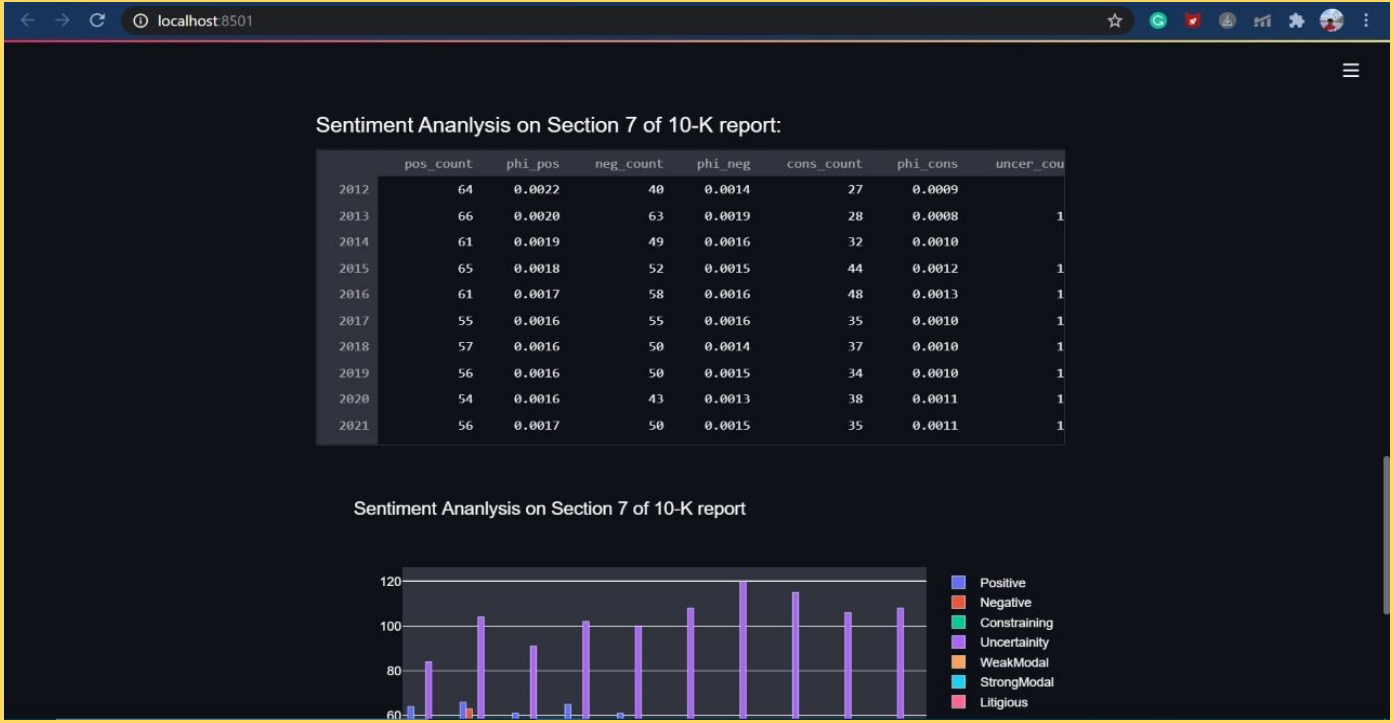


	pos_count	phi_pos	neg_count	phi_neg	cons_count	phi_cons	uncer_cou
2012	66	0.0029	208	0.0091	55	0.0024	1
2013	66	0.0029	209	0.0091	55	0.0024	1
2014	65	0.0028	204	0.0089	54	0.0024	1
2015	68	0.0027	217	0.0087	65	0.0026	2
2016	69	0.0027	218	0.0086	65	0.0026	2
2017	69	0.0027	219	0.0086	70	0.0027	2
2018	70	0.0026	221	0.0082	70	0.0026	2
2019	68	0.0027	228	0.0089	66	0.0026	2
2020	71	0.0025	258	0.0091	76	0.0027	1
2021	71	0.0025	254	0.0090	74	0.0026	1

Final Project:

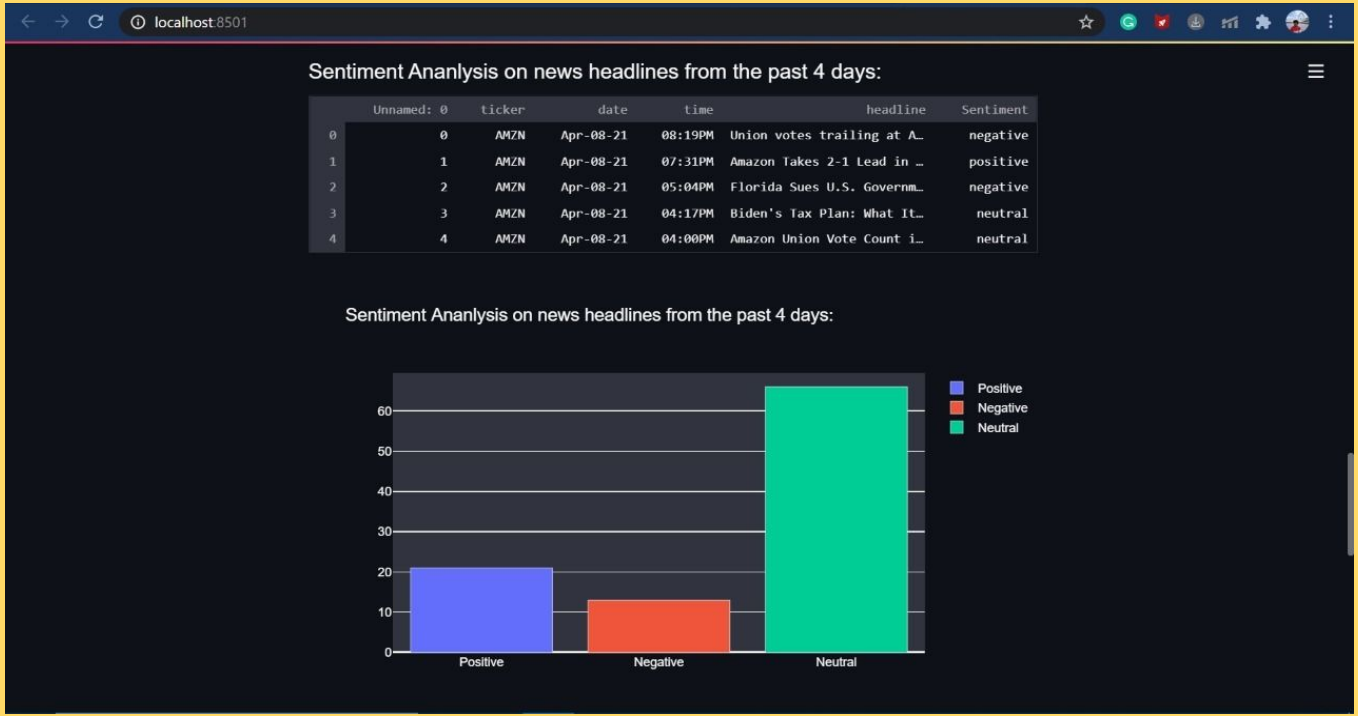
2017	69	0.0027	219	0.0086	70	0.0027	2
2018	70	0.0026	221	0.0082	70	0.0026	2
2019	68	0.0027	228	0.0089	66	0.0026	2
2020	71	0.0025	258	0.0091	76	0.0027	1
2021	71	0.0025	254	0.0090	74	0.0026	1

Final Project:



	pos_count	phi_pos	neg_count	phi_neg	cons_count	phi_cons	uncer_cou
2012	64	0.0022	40	0.0014	27	0.0009	
2013	66	0.0020	63	0.0019	28	0.0008	1
2014	61	0.0019	49	0.0016	32	0.0010	
2015	65	0.0018	52	0.0015	44	0.0012	1
2016	61	0.0017	58	0.0016	48	0.0013	1
2017	55	0.0016	55	0.0016	35	0.0010	1
2018	57	0.0016	50	0.0014	37	0.0010	1
2019	56	0.0016	50	0.0015	34	0.0010	1
2020	54	0.0016	43	0.0013	38	0.0011	1
2021	56	0.0017	50	0.0015	35	0.0011	1

Final Project:



	Unnamed: 0	ticker	date	time	headline	Sentiment
0	0	AMZN	Apr-08-21	08:19PM	Union votes trailing at A…	negative
1	1	AMZN	Apr-08-21	07:31PM	Amazon Takes 2-1 Lead in …	positive
2	2	AMZN	Apr-08-21	05:04PM	Florida Sues U.S. Governm…	negative
3	3	AMZN	Apr-08-21	04:17PM	Biden's Tax Plan: What It…	neutral
4	4	AMZN	Apr-08-21	04:00PM	Amazon Union Vote Count i…	neutral

Thank You.