Sentiment Analysis of Federal Reserve Announcements and its Impact on the Sector ETFs

Sz-Je Wang, Sikai Wang, Xilong Liu

Overview

1. Research Question and Hypothesis

2. Necessary Data

3. How we load and analyze data

Research Question

How do Federal Reserve announcements influence the stock market according to Natural language processing, particularly the ETFs across various sectors?

Specific Research Question

- What is the overall sentiment in the each announcements?
- How do these sentiment scores correlate with subsequent stock market movements for the sector ETFs?
- Can we predict stock market reactions based on the sentiment derived from announcements?

SELECT SECTOR SPDRs®

Hypothesis

- Positive sentiment score in announcements leads to a positive stock market reaction.

- Negative sentiment score in announcements leads to a negative stock market reaction.

Predictions

According to the hypotheses above, we believe that every time when the Federal Reserve announcements related to a piece of good news are released, the stock of companies will show a good trend and vice versa. Whether it has a piece of good news is determined by the sentiment score that we measure.

Necessary Data

Federal Reserve Board

- **Observation** Federal Reserve announcements and sector ETFs.
- **Sample Period** Recent 10 most significant Federal Reserve announcements.
- **Necessary Variables** Sentiment scores, stock prices, and returns.

1/10/2023	*Panel on "Central Bank Independence and the Mandate—Evolving Views"* **Watch Live** Chair Jerome H. Powell At the Symposium on Central Bank Independence, Sveriges Riksbank, Stockholm, Sweden
11/30/2022	*Inflation and the Labor Market* **Watch Live** Chair Jerome H. Powell At the Hutchins Center on Fiscal and Monetary Policy, Brookings Institution, Washington, D.C.

Sentiment list

Positive word	Negative word
benefited	underperformance
continue	goodwill
outperformance	decreased
increased	trying
excellent	resolve
growth	term
increase	believe
driving	experienced
helping	slipped
drove	slowed
grew	longer
performance	affecting
	declines

Yahoo Finance

Financials: Financial Select Sector SPDR Fund (XLF)
Technology: Technology Select Sector SPDR Fund (XLK)
Healthcare: Health Care Select Sector SPDR Fund (XLV)
Consumer Discretionary: Consumer Discretionary Select Sector SPDR Fund (XLY)
Consumer Staples: Consumer Staples Select Sector SPDR Fund (XLP)
Industrials: Industrial Select Sector SPDR Fund (XLI)
Energy: Energy Select Sector SPDR Fund (XLE)
Materials: Materials Select Sector SPDR Fund (XLB)
Utilities: Utilities Select Sector SPDR Fund (XLU)
Real Estate: Real Estate Select Sector SPDR Fund (XLRE)
Communication Services: Communication Services Select Sector SPDR Fund (XLC)

How we Load and Analyze data

1. Visualization

2. Sentiment Score

3. Analysis

Visualization.ipynb

Merge Overall Data

	Ticker	Date	Open	High	Low	Close	Adj Close	Volume	Return
0	XLB	2021/9/7	85.070000	85.129997	84.690002	84.809998	81.769379	6932300	NaN
1	XLB	2021/9/8	84.559998	84.639999	83.709999	84.000000	80.988419	6210000	-0.0100
2	XLB	2021/9/9	83.760002	84.599998	83.760002	84.029999	81.017342	4935700	0.0000
3	XLB	2021/9/10	84.419998	85.050003	83.959999	84.010002	80.998062	8890100	0.0000
4	XLB	2021/9/13	84.750000	85.000000	83.459999	83.989998	80.978775	7610000	0.0000
...	...	...	...	...	...	...	...	...	...
3834	XLC	2023/1/19	50.970001	51.755001	50.799999	51.490002	51.349598	7387400	0.0051
3835	XLC	2023/1/20	51.910000	53.155998	51.770000	53.099998	52.955204	4874200	0.0313
3836	XLC	2023/1/23	53.150002	54.154999	53.020000	54.029999	53.882671	6279000	0.0175
3837	XLC	2023/1/24	53.580002	54.650002	53.412998	53.880001	53.733082	8867000	-0.0028
3838	XLC	2023/1/25	53.500000	54.189999	53.310001	54.070000	53.922562	4976800	0.0035

3839 rows × 9 columns

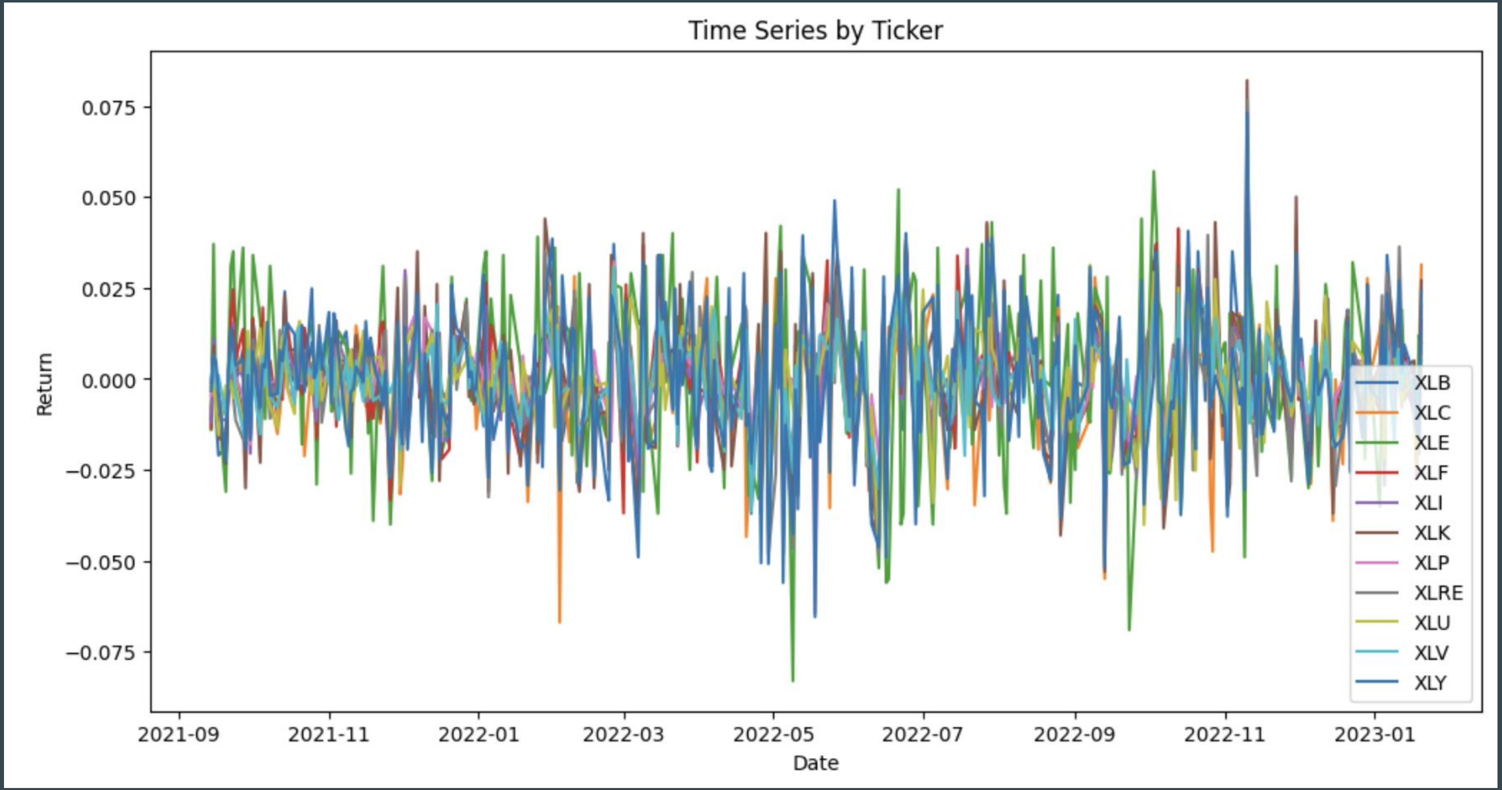
```python
# Define the events
event_dates = {'Event 1': ['2021/09/14', '2021/10/04'],
               'Event 2': ['2021/10/30', '2021/11/19'],
               'Event 3': ['2021/11/19', '2021/12/09'],
               'Event 4': ['2022/03/11', '2022/03/31'],
               'Event 5': ['2022/05/14', '2022/06/03'],
               'Event 6': ['2022/06/07', '2022/06/27'],
               'Event 7': ['2022/08/16', '2022/09/05'],
               'Event 8': ['2022/09/18', '2022/10/08'],
               'Event 9': ['2022/11/20', '2022/12/10'],
               'Event 10': ['2023/01/01', '2023/01/21']}

# Convert date strings to datetime objects
for event, dates in event_dates.items():
    event_dates[event] = [pd.to_datetime(date) for date in dates]

# Add Event column to returns DataFrame
returns['Event'] = ''
for event, dates in event_dates.items():
    start_date, end_date = dates
    mask = (returns['Date'] >= start_date - pd.Timedelta(days=10)) & \
           (returns['Date'] <= end_date + pd.Timedelta(days=10))
    returns.loc[mask, 'Event'] = event

returns
returns.to_csv('analysis csv file/returns.csv', index=False)
```

Cumulative excess return for that asset during that event's time window (from t-10 up to t+10)

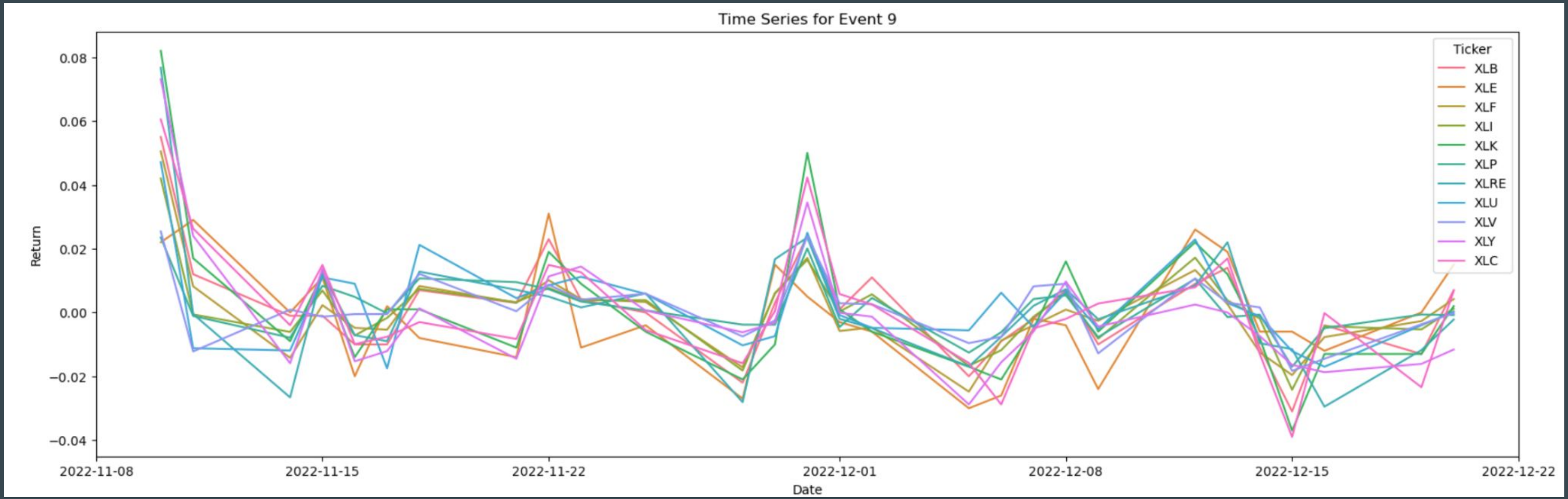
Output

	Ticker	Date	Return	Event
0	XLB	2021-09-14	-0.0110	Event 1
1	XLE	2021-09-14	-0.0140	Event 1
2	XLF	2021-09-14	-0.0134	Event 1
3	XLI	2021-09-14	-0.0120	Event 1
4	XLK	2021-09-14	-0.0010	Event 1
...	...	...	...	...
3746	XLRE	2023-01-20	0.0116	Event 10
3747	XLU	2023-01-20	0.0060	Event 10
3748	XLV	2023-01-20	0.0050	Event 10
3749	XLY	2023-01-20	0.0242	Event 10
3750	XLC	2023-01-20	0.0313	Event 10

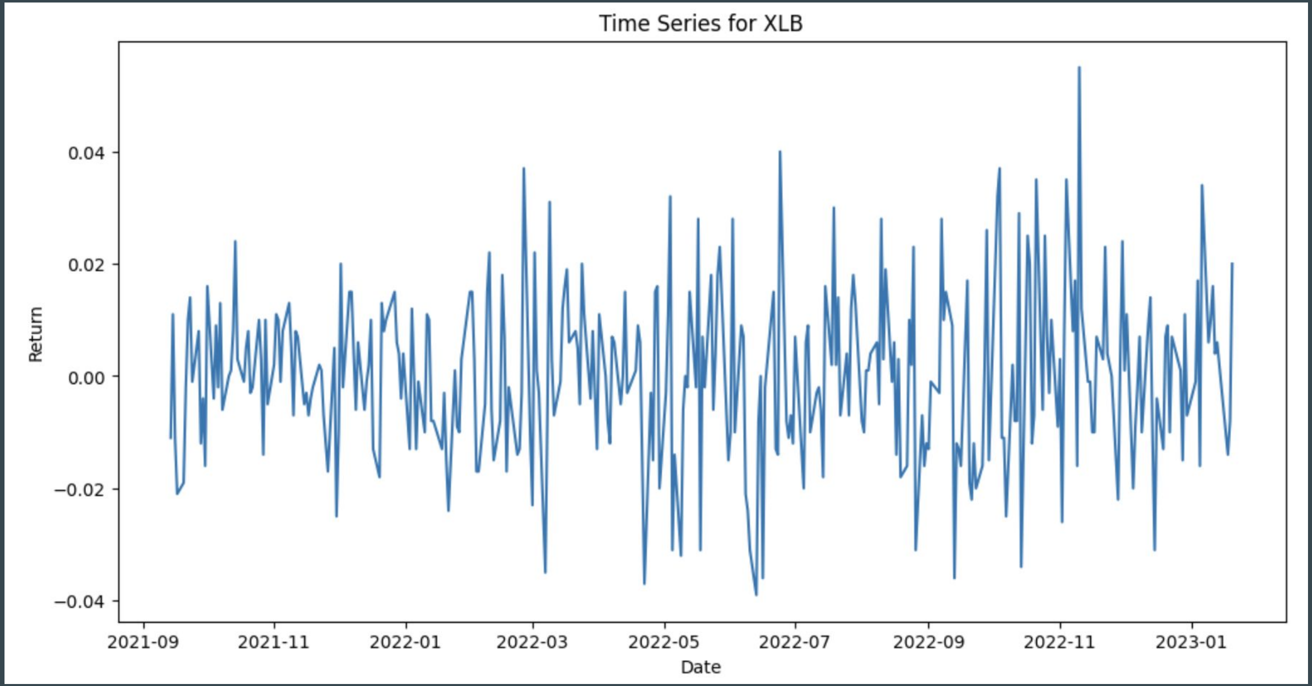
Time Series for all Ticker from 2021.9 to 2023.1 (10 announcement period) (most beautiful one)



Event 9 (2022.11.30) Fed Announcement



Time series for XLB (Materials)



Sentiment Analysis.ipynb

```
fed_20210924 = PyPDF2.PdfReader('Fed/20210924.pdf')
fed_20211109 = PyPDF2.PdfReader('Fed/20211109.pdf')
fed_20211129 = PyPDF2.PdfReader('Fed/20211129.pdf')
fed_20220321 = PyPDF2.PdfReader('Fed/20220321.pdf')
fed_20220524 = PyPDF2.PdfReader('Fed/20220524.pdf')
fed_20220617 = PyPDF2.PdfReader('Fed/20220617.pdf')
fed_20220826 = PyPDF2.PdfReader('Fed/20220826.pdf')
fed_20220928 = PyPDF2.PdfReader('Fed/20220928.pdf')
fed_20221130 = PyPDF2.PdfReader('Fed/20221130.pdf')
fed_20230110 = PyPDF2.PdfReader('Fed/20230110.pdf')
```

	Date	SentimentScore
0	2021-09-24	8
1	2021-11-09	26
2	2021-11-29	21
3	2022-03-21	2
4	2022-05-24	28
5	2022-06-17	9
6	2022-08-26	18
7	2022-09-28	12
8	2022-11-30	28
9	2023-01-10	1

```
positive_words = load_sentiment_words('Sentiment/ML_positive_unigram.txt')
negative_words = load_sentiment_words('Sentiment/ML_negative_unigram.txt')

# Calculate sentiment score using custom sentiment word lists
def custom_sentiment_score(text):
    tokenized_text = word_tokenize(text)
    positive_count = sum([1 for word in tokenized_text if word.lower() in positive_words])
    negative_count = sum([1 for word in tokenized_text if word.lower() in negative_words])
    return positive_count - negative_count
```

Analysis.ipynb

Returns

	Ticker	Date	Return	Event
0	XLB	2021-09-14	-0.0110	Event 1
1	XLE	2021-09-14	-0.0140	Event 1
2	XLF	2021-09-14	-0.0134	Event 1
3	XLI	2021-09-14	-0.0120	Event 1
4	XLK	2021-09-14	-0.0010	Event 1
...	...	...	...	...
3746	XLRE	2023-01-20	0.0116	Event 10
3747	XLU	2023-01-20	0.0060	Event 10
3748	XLV	2023-01-20	0.0050	Event 10
3749	XLY	2023-01-20	0.0242	Event 10
3750	XLC	2023-01-20	0.0313	Event 10

3751 rows × 4 columns

	Date	SentimentScore
0	2021-09-24	8
1	2021-11-09	26
2	2021-11-29	21
3	2022-03-21	2
4	2022-05-24	28
5	2022-06-17	9
6	2022-08-26	18
7	2022-09-28	12
8	2022-11-30	28
9	2023-01-10	1

Sentiment Score

Code And Output

```python
window = 10
event_dates = ['2021-09-24', '2021-11-09', '2021-11-29', '2022-03-21', '2022-05-24', '2022-06-17', '2022-08-26', '2022-09-28', '2022-11-30', '2023-01-11']
event_dates = [pd.Timestamp(date) for date in event_dates]
returns['Date'] = pd.to_datetime(returns['Date'])

for event_date in event_dates:
    before_event = returns[returns['Date'] < event_date].tail(window)
    after_event = returns[returns['Date'] > event_date].head(window)

    mean_before = before_event['Return'].mean()
    std_before = before_event['Return'].std()
    mean_after = after_event['Return'].mean()
    std_after = after_event['Return'].std()

    return_mean = (mean_before-mean_after)/mean_before
    return_std = (std_before-std_after)/std_before
    announce_date.loc[announce_date['Date'] == event_date, 'Return_mean'] = return_mean
    announce_date.loc[announce_date['Date'] == event_date, 'Return_Standard_Deviation'] = return_std

sentiment_score_df['Date'] = pd.to_datetime(sentiment_score_df['Date'])
merged_df = pd.merge(announce_date, sentiment_score_df, on='Date')
print(merged_df)

corr_matrix = merged_df[['Return_mean', 'Return_Standard_Deviation', 'SentimentScore']].corr()
print(corr_matrix)
```

```
         Date  Return_mean  Return_Standard_Deviation  SentimentScore
0 2021-09-24     0.936924                  -0.298629               8
1 2021-11-09    -3.196928                  -0.136439              26
2 2021-11-29     0.069060                   0.400765              21
3 2022-03-21    -0.106188                   0.005310               2
4 2022-05-24     0.520890                  -0.200474              28
5 2022-06-17     1.815041                   0.313768               9
6 2022-08-26     1.290273                  -0.577637              18
7 2022-09-28    -3.113639                  -0.364481              12
8 2022-11-30     1.642191                   0.699407              28
                           Return_mean  Return_Standard_Deviation  \
Return_mean                   1.000000                   0.349404
Return_Standard_Deviation     0.349404                   1.000000
SentimentScore               -0.063444                   0.227383

                           SentimentScore
Return_mean                     -0.063444
Return_Standard_Deviation        0.227383
SentimentScore                   1.000000
```

Market Return!!!!

We regard 10 days before the announcement as before_event and 10 days after the announcement as after_event. Find out the return on their means and std for each announcement.

It seems like the relationship between them is not strong enough. And we then apply this mean and STD for each ETF.

FINAL RESULT

	Ticker	Metric	Return_mean_corr	Return_std_corr	SentimentScore_corr
0	XLB	Return_mean	1.000000	-0.250130	-0.411511
1	XLB	Return_std	-0.250130	1.000000	-0.160990
2	XLB	SentimentScore	-0.411511	-0.160990	1.000000
3	XLC	Return_mean	1.000000	-0.323311	-0.033343
4	XLC	Return_std	-0.323311	1.000000	-0.168661
5	XLC	SentimentScore	-0.033343	-0.168661	1.000000
6	XLE	Return_mean	1.000000	-0.015576	-0.451910
7	XLE	Return_std	-0.015576	1.000000	-0.392416
8	XLE	SentimentScore	-0.451910	-0.392416	1.000000

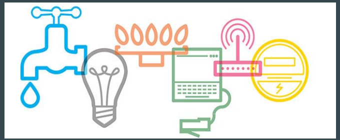
Correlation for each ETF According to Mean, STD, and Sentiment Score

Finding 2022-11-30

- Possibility of slowing down interest rate hikes as early as the December meeting.

- Consider the duration of maintaining restrictive interest rates to curb inflation.

Utilities



XLU	2022-11-30	5.444444e-01	0.253182	28

Materials

XLB	2022-11-30	-2.333333	0.060507	28

Financials

XLF	2022-11-30	5.734694	-0.235137	28

Healthcare

XLV	2022-11-30	5.691489e-01	-0.405344	28

Our Website

https://sikaiwang224.github.io/teamproject-/

Q&A

THANK YOU