Modern NLP Approach - Transformer-Based Sentiment Analysis

Load The Dataset

```
# Mount Google Drive (if using Colab)
from google.colab import drive
drive.mount('/content/drive')

import pandas as pd

# Load cleaned dataset
input_path = "/content/drive/My Drive/NLP/Assignment_3/fomc_transcripts_spacy_cleaned.csv
df = pd.read_csv(input_path)
df['Date'] = pd.to_datetime(df['Date'])
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m
```

Install required libraries

```
!pip install --upgrade transformers
```

```
Requirement already satisfied: transformers in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (f
Requirement already satisfied: huggingface-hub<1.0,>=0.30.0 in /usr/local/lib/python3
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (f
Requirement already satisfied: tokenizers<0.22,>=0.21 in /usr/local/lib/python3.11/di
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.1
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-p
```

```
!pip install --upgrade torch torchvision torchaudio
```

```
Requirement already satisfied: torch in /usr/local/lib/python3.11/dist-packages (2.6.
```

```
Collecting torch
  Downloading torch-2.7.0-cp311-cp311-manylinux_2_28_x86_64.whl.metadata (29 kB)
Requirement already satisfied: torchvision in /usr/local/lib/python3.11/dist-packages
Collecting torchvision
  Downloading torchvision-0.22.0-cp311-cp311-manylinux_2_28_x86_64.whl.metadata (6.1
Requirement already satisfied: torchaudio in /usr/local/lib/python3.11/dist-packages
Collecting torchaudio
  Downloading torchaudio-2.7.0-cp311-cp311-manylinux_2_28_x86_64.whl.metadata (6.6 kB
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (f
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.11
Collecting sympy>=1.13.3 (from torch)
  Downloading sympy-1.14.0-py3-none-any.whl.metadata (12 kB)
Requirement already satisfied: networkx in /usr/local/lib/python3.11/dist-packages (f
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packages (fro
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packages (fro
Collecting nvidia-cuda-nvrtc-cu12==12.6.77 (from torch)
  Downloading nvidia_cuda_nvrtc_cu12-12.6.77-py3-none-manylinux2014_x86_64.whl.metada
Collecting nvidia-cuda-runtime-cu12==12.6.77 (from torch)
  Downloading nvidia_cuda_runtime_cu12-12.6.77-py3-none-manylinux2014_x86_64.manylinu
Collecting nvidia-cuda-cupti-cu12==12.6.80 (from torch)
  Downloading nvidia_cuda_cupti_cu12-12.6.80-py3-none-manylinux2014_x86_64.manylinux_
Collecting nvidia-cudnn-cu12==9.5.1.17 (from torch)
  Downloading nvidia_cudnn_cu12-9.5.1.17-py3-none-manylinux_2_28_x86_64.whl.metadata
Collecting nvidia-cublas-cu12==12.6.4.1 (from torch)
  Downloading nvidia_cublas_cu12-12.6.4.1-py3-none-manylinux2014_x86_64.manylinux_2_1
Collecting nvidia-cufft-cu12==11.3.0.4 (from torch)
  Downloading nvidia_cufft_cu12-11.3.0.4-py3-none-manylinux2014_x86_64.manylinux_2_17
Collecting nvidia-curand-cu12==10.3.7.77 (from torch)
  Downloading nvidia_curand_cu12-10.3.7.77-py3-none-manylinux2014_x86_64.manylinux_2_
Collecting nvidia-cusolver-cu12==11.7.1.2 (from torch)
  Downloading nvidia_cusolver_cu12-11.7.1.2-py3-none-manylinux2014_x86_64.manylinux_2
Collecting nvidia-cusparse-cu12==12.5.4.2 (from torch)
  Downloading nvidia_cusparse_cu12-12.5.4.2-py3-none-manylinux2014_x86_64.manylinux_2
Collecting nvidia-cusparselt-cu12==0.6.3 (from torch)
  Downloading nvidia_cusparselt_cu12-0.6.3-py3-none-manylinux2014_x86_64.whl.metadata
Collecting nvidia-nccl-cu12==2.26.2 (from torch)
  Downloading nvidia_nccl_cu12-2.26.2-py3-none-manylinux2014_x86_64.manylinux_2_17_x8
Collecting nvidia-nvtx-cu12==12.6.77 (from torch)
  Downloading nvidia_nvtx_cu12-12.6.77-py3-none-manylinux2014_x86_64.manylinux_2_17_x
Collecting nvidia-nvjitlink-cu12==12.6.85 (from torch)
  Downloading nvidia_nvjitlink_cu12-12.6.85-py3-none-manylinux2010_x86_64.manylinux_2
Collecting nvidia-cufile-cu12==1.11.1.6 (from torch)
  Downloading nvidia_cufile_cu12-1.11.1.6-py3-none-manylinux2014_x86_64.manylinux_2_1
Collecting triton==3.3.0 (from torch)
  Downloading triton-3.3.0-cp311-cp311-manylinux_2_27_x86_64.manylinux_2_28_x86_64.wh
Requirement already satisfied: setuptools>=40.8.0 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in /usr/local/lib/python3.11/dis
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-pack
Downloading torch-2.7.0-cp311-cp311-manylinux_2_28_x86_64.whl (865.2 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 865.2/865.2 MB 1.8 MB/s eta 0:00:00
Downloading nvidia_cublas_cu12-12.6.4.1-py3-none-manylinux2014_x86_64.manylinux_2_17_
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 393.1/393.1 MB 1.8 MB/s eta 0:00:00
Downloading nvidia_cuda_cupti_cu12-12.6.80-py3-none-manylinux2014_x86_64.manylinux_2_
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.9/8.9 MB 76.5 MB/s eta 0:00:00
     Downloading nvidia_cuda_nvrtc_cu12-12.6.77-py3-none-manylinux2014_x86_64.whl (23.7 MB
```

```
import torch
print(torch.__version__)
```

```
     2.7.0+cu126
```

```
!pip install transformers --upgrade --user
```

```
     Requirement already satisfied: transformers in /usr/local/lib/python3.11/dist-package
     Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (f
     Requirement already satisfied: huggingface-hub<1.0,>=0.30.0 in /usr/local/lib/python3
     Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-packages
     Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-pack
     Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-packages
     Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.11/dist-pa
     Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (f
     Requirement already satisfied: tokenizers<0.22,>=0.21 in /usr/local/lib/python3.11/di
     Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.11/dist-p
     Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.11/dist-packages
     Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.11/dist-pac
     Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.1
     Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/
     Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-package
     Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-p
     Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-p
```

```
import pandas as pd
from transformers import AutoTokenizer, AutoModelForSequenceClassification, pipeline
from collections import Counter  # Import Counter to compute the most frequent label

# Load the FinBERT model from Hugging Face
model_name = "ProsusAI/finbert"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSequenceClassification.from_pretrained(model_name)

# Create a sentiment-analysis pipeline using FinBERT
finbert_pipeline = pipeline("sentiment-analysis", model=model, tokenizer=tokenizer)
```

```
     tokenizer_config.json: 100%                                  252/252 [00:00<00:00, 7.03kB/
                                                                  s]
     config.json: 100%                                            758/758 [00:00<00:00, 33.5kB/s]
     vocab.txt: 100%                                              232k/232k [00:00<00:00, 4.22MB/s]
     special_tokens_map.json: 100%                                112/112 [00:00<00:00, 4.21kB/
                                                                  s]
     Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Fall
     WARNING:huggingface_hub.file_download:Xet Storage is enabled for this repo, but the '
```

```
WARNING:huggingface_hub.file_download:Xet Storage is enabled for this repo, but the
```

```
pytorch_model.bin: 100%  438M/438M [00:07<00:00, 134MB/s]
```

FinBERT Sentiment Analysis

Sentiment Scoring using FINBERT

```
!pip install spacy
import spacy
import pandas as pd
from transformers import pipeline, AutoTokenizer
from collections import defaultdict

# Load SpaCy model
nlp = spacy.load("en_core_web_sm")

# Load FinBERT tokenizer and sentiment pipeline with updated arguments
tokenizer = AutoTokenizer.from_pretrained("ProsusAI/finbert")
finbert_pipeline = pipeline(
    "sentiment-analysis",
    model="ProsusAI/finbert",
    tokenizer=tokenizer,
    top_k=None  # Return all scores per sentiment
)

# Function to create sliding chunks (~450 tokens per chunk with ~200 token overlap)
def create_sliding_chunks(text, tokenizer, max_tokens=450, stride=200):
    doc = nlp(text)
    sentences = [sent.text.strip() for sent in doc.sents]
    sentence_lengths = [len(tokenizer.encode(sent, add_special_tokens=False)) for sent in

    chunks = []
    start = 0
    while start < len(sentences):
        token_count = 0
        end = start
        while end < len(sentences) and token_count + sentence_lengths[end] <= max_tokens:
            token_count += sentence_lengths[end]
            end += 1
        chunk = " ".join(sentences[start:end])
        chunks.append(chunk)

        # Move start index forward by approximately stride tokens
        moved = 0
        while start < len(sentences) and moved < stride:
            moved += sentence_lengths[start]
```

```
            start += 1
    return chunks

# Function to calculate sentiment score using Positive - Negative formula
def get_sentiment_pos_minus_neg(text, tokenizer, finbert_pipeline):
    chunks = create_sliding_chunks(text, tokenizer)
    sentiment_scores = defaultdict(list)

    for chunk in chunks:
        # Run FinBERT sentiment model on chunk with truncation protection
        scores = finbert_pipeline(chunk, truncation=True, max_length=512)[0]
        for entry in scores:
            label = entry['label'].upper()
            sentiment_scores[label].append(entry['score'])

    # Calculate average scores
    avg_pos = sum(sentiment_scores['POSITIVE']) / len(sentiment_scores['POSITIVE']) if se
    avg_neg = sum(sentiment_scores['NEGATIVE']) / len(sentiment_scores['NEGATIVE']) if se

    sentiment_score = round(avg_pos - avg_neg, 4)
    return pd.Series({"sentiment_score": sentiment_score})

# Apply sentiment logic to DataFrame
df[['sentiment_score']] = df['cleaned_text'].apply(
    lambda x: get_sentiment_pos_minus_neg(x, tokenizer, finbert_pipeline)
)
```

```
Requirement already satisfied: spacy in /usr/local/lib/python3.11/dist-packages (3.8.
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.11
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.11/dis
Requirement already satisfied: thinc<8.4.0,>=8.3.4 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.11/dist
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.11/d
Requirement already satisfied: weasel<0.5.0,>=0.1.0 in /usr/local/lib/python3.11/dist
Requirement already satisfied: typer<1.0.0,>=0.3.0 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: numpy>=1.19.0 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/d
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packages (fro
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in /usr/local/lib/python3.11/d
Requirement already satisfied: language-data>=1.2 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.11/di
Requirement already satisfied: pydantic-core==2.33.1 in /usr/local/lib/python3.11/dis
Requirement already satisfied: typing-extensions>=4.12.2 in /usr/local/lib/python3.11
```

```
Requirement already satisfied: typing-extensions>=4.12.2 in /usr/local/lib/python3.11
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.11/
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: blis<1.4.0,>=1.3.0 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.11/
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.1
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.11/
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: marisa-trie>=1.1.0 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dis
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/d
Requirement already satisfied: wrapt in /usr/local/lib/python3.11/dist-packages (from
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-packages
Device set to use cpu
Token indices sequence length is longer than the specified maximum sequence length fo
```

```
df
```

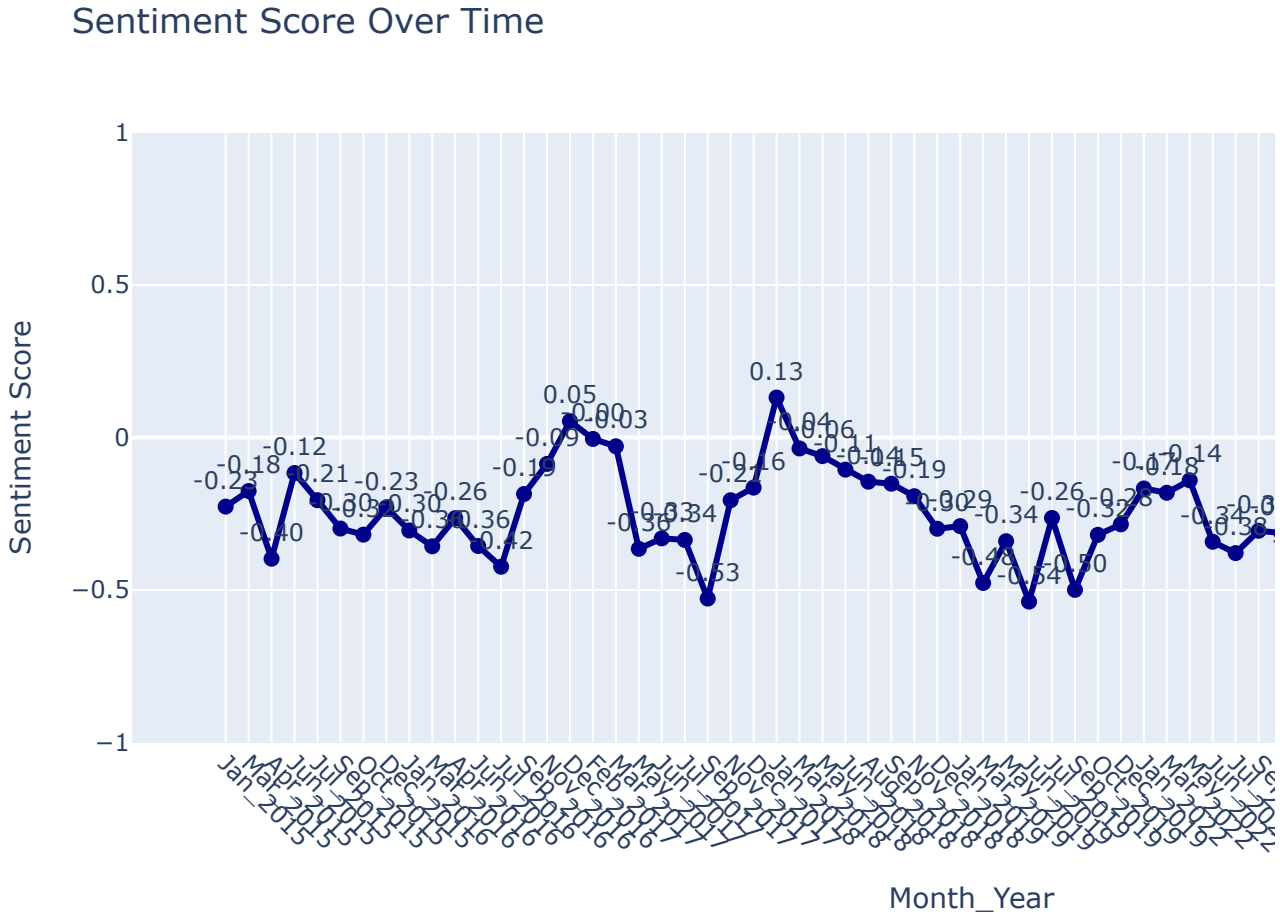
	URL	Date	Year	Month	Day	Content	date
0	https://www.federalreserve.gov/monetarypolicy/...	2025-03-19	2025	2025-03-01	19	HomeMonetary PolicyFederal Open Market Committ...	2025-03-31
1	https://www.federalreserve.gov/monetarypolicy/...	2025-01-29	2025	2025-01-01	29	HomeMonetary PolicyFederal Open Market Committ...	2025-01-31
2	https://www.federalreserve.gov/monetarypolicy/...	2024-12-18	2024	2024-12-01	18	HomeMonetary PolicyFederal Open Market Committ...	2024-12-31
3	https://www.federalreserve.gov/monetarypolicy/...	2024-11-07	2024	2024-11-01	7	HomeMonetary PolicyFederal Open Market Committ...	2024-11-30
4	https://www.federalreserve.gov/monetarypolicy/...	2024-09-18	2024	2024-09-01	18	HomeMonetary PolicyFederal Open Market Committ...	2024-09-30
...	...	...	...	...	...	...	...
61	https://www.federalreserve.gov/	2015-07-29	2015	2015-07-01	29	HomeMonetary PolicyFederal Open Market	2015-07-31

	monetarypolicy/...					Open Market Committ...	
62	https://www.federalreserve.gov/monetarypolicy/...	2015-06-17	2015	2015-06-01	17	HomeMonetary PolicyFederal Open Market Committ...	2015-06-30
63	https://www.federalreserve.gov/monetarypolicy/...	2015-04-29	2015	2015-04-01	29	HomeMonetary PolicyFederal Open Market Committ...	2015-04-30
64	https://www.federalreserve.gov/monetarypolicy/...	2015-03-18	2015	2015-03-01	18	HomeMonetary PolicyFederal Open Market Committ...	2015-03-31
65	https://www.federalreserve.gov/monetarypolicy/...	2015-01-28	2015	2015-01-01	28	HomeMonetary PolicyFederal Open Market Committ...	2015-01-31

66 rows × 46 columns

```
import pandas as pd
import plotly.graph_objs as go

# Ensure Date is datetime and sorted
df['Date'] = pd.to_datetime(df['Date'])
df = df.sort_values(by='Date')

# Create a formatted string for Month_Year to use on the x-axis
df['Month_Year'] = df['Date'].dt.strftime('%b_%Y')  # e.g., Jan_2024

# Create line plot of sentiment scores
fig = go.Figure()

fig.add_trace(go.Scatter(
    x=df['Month_Year'],
    y=df['sentiment_score'],
    mode='lines+markers+text',
    text=[f"{score:.2f}" for score in df['sentiment_score']],
    textposition="top center",
    name="Sentiment Score",
    line=dict(color='darkblue', width=3),
    marker=dict(size=8)
))

# Update layout
fig.update_layout(
    title="Sentiment Score Over Time",
```

```
    xaxis_title="Month_Year",
    yaxis_title="Sentiment Score",
    xaxis=dict(
        tickmode='array',
        tickvals=df['Month_Year'],
        ticktext=df['Month_Year'],
        tickangle=45
    ),
    yaxis=dict(range=[-1, 1]),
    height=500,
    width=1000
)

fig.show()
```

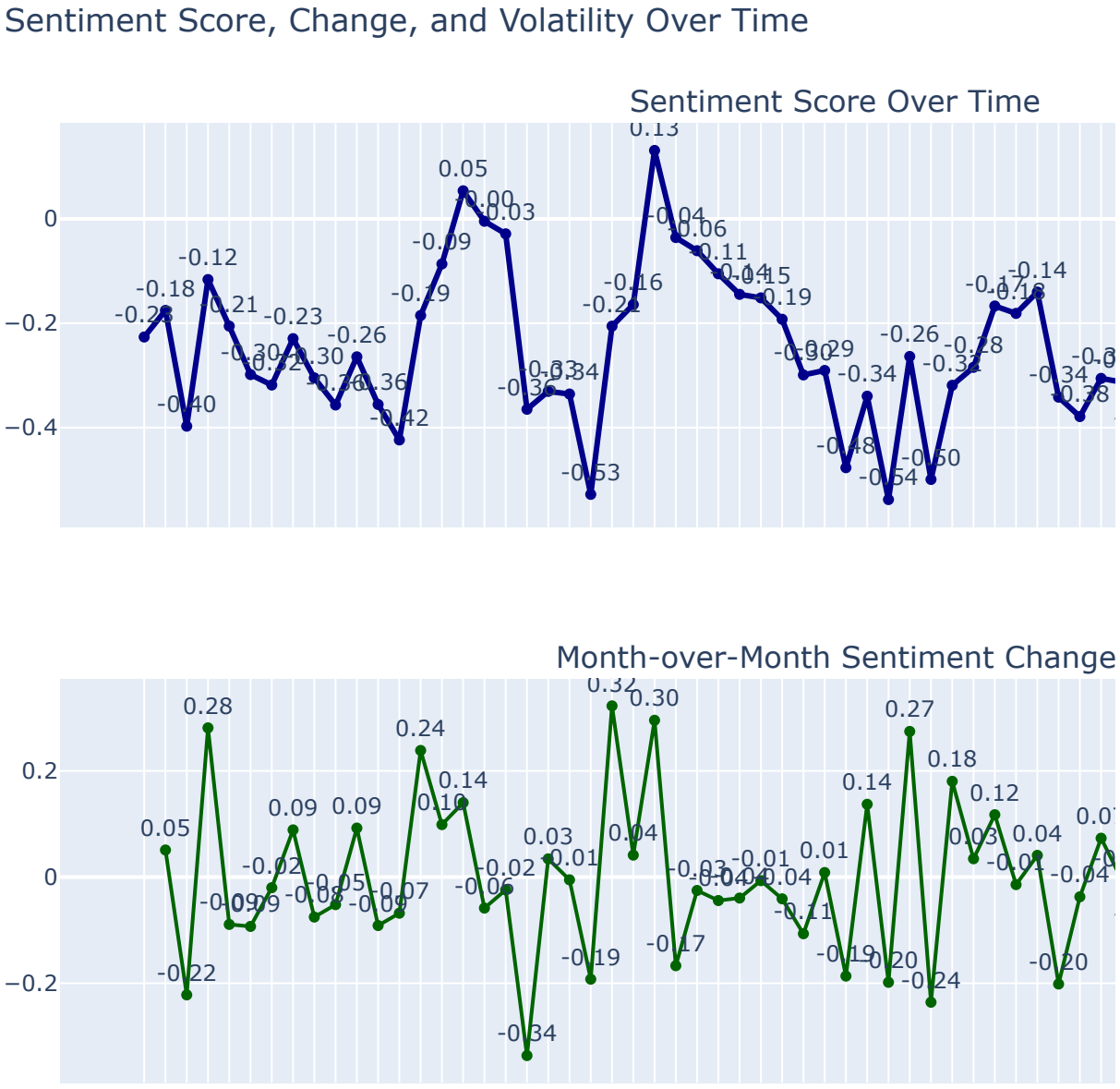


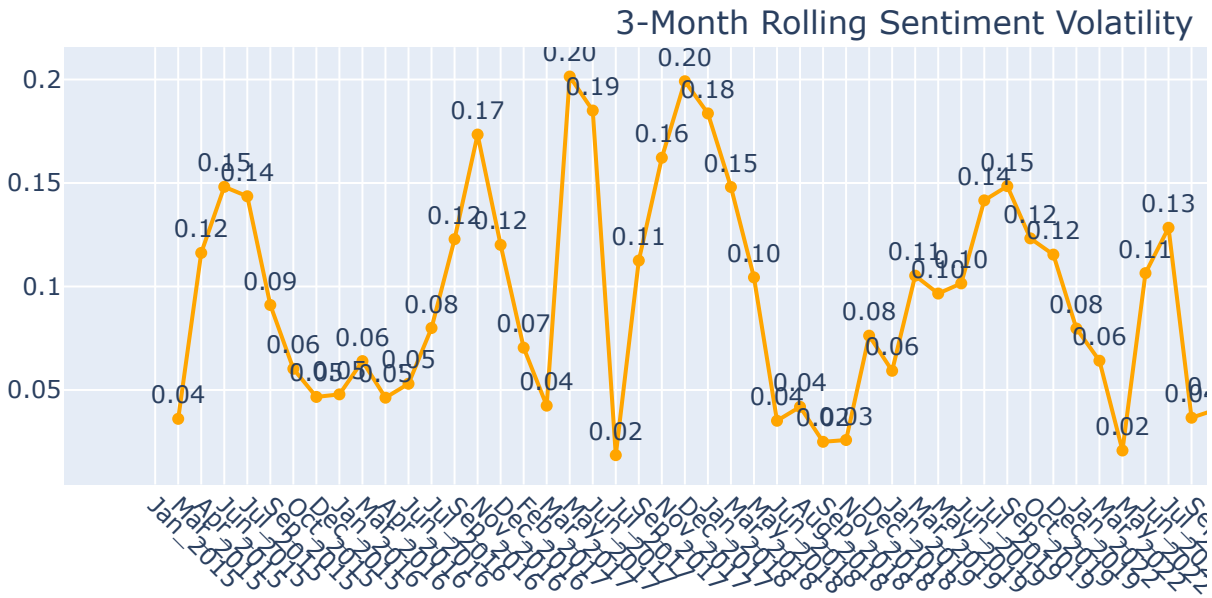

```
df_1 = df.copy()
```

```
import pandas as pd
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

```
# Work with df_1 instead of df, and avoid overwriting original Date
df_1 = df_1.sort_values(by='Date')  # Sort chronologically
df_1['date_new'] = df_1['Date'].dt.strftime('%b_%Y')  # New column for formatted dates

# Create sentiment trend metrics
df_1['sentiment_delta'] = df_1['sentiment_score'].diff()
df_1['sentiment_volatility'] = df_1['sentiment_score'].rolling(window=3, min_periods=1).s

# Build subplot layout
fig = make_subplots(
    rows=3, cols=1,
    shared_xaxes=True,
    vertical_spacing=0.1,
    subplot_titles=[
        "Sentiment Score Over Time",
        "Month-over-Month Sentiment Change",
        "3-Month Rolling Sentiment Volatility"
    ]
)

# Plot 1: Sentiment Score
fig.add_trace(go.Scatter(
    x=df_1['date_new'],
    y=df_1['sentiment_score'],
    mode='lines+markers+text',
    text=[f"{v:.2f}" for v in df_1['sentiment_score']],
    textposition="top center",
    name="Sentiment Score",
    line=dict(color='darkblue', width=3)
), row=1, col=1)

# Plot 2: Month-over-Month Change
fig.add_trace(go.Scatter(
    x=df_1['date_new'],
    y=df_1['sentiment_delta'],
    mode='lines+markers+text',
    text=[f"{v:.2f}" if pd.notna(v) else "" for v in df_1['sentiment_delta']],
    textposition="top center",
    name="Sentiment Change",
    line=dict(color='darkgreen')
), row=2, col=1)

# Plot 3: Volatility (Rolling Std Dev)
fig.add_trace(go.Scatter(
    x=df_1['date_new'],
    y=df_1['sentiment_volatility'],
    mode='lines+markers+text',
    text=[f"{v:.2f}" if pd.notna(v) else "" for v in df_1['sentiment_volatility']],
    textposition="top center",
    name="Volatility (3-Month Std Dev)"
```

```
    name='Volatility (3-Month Std Dev)',
    line=dict(color='orange')
), row=3, col=1)

# Final layout settings
fig.update_layout(
    height=1000,
    width=1000,
    title="Sentiment Score, Change, and Volatility Over Time",
    showlegend=False
)

fig.update_xaxes(tickangle=45, tickmode='array', tickvals=df_1['date_new'])

fig.show()
```



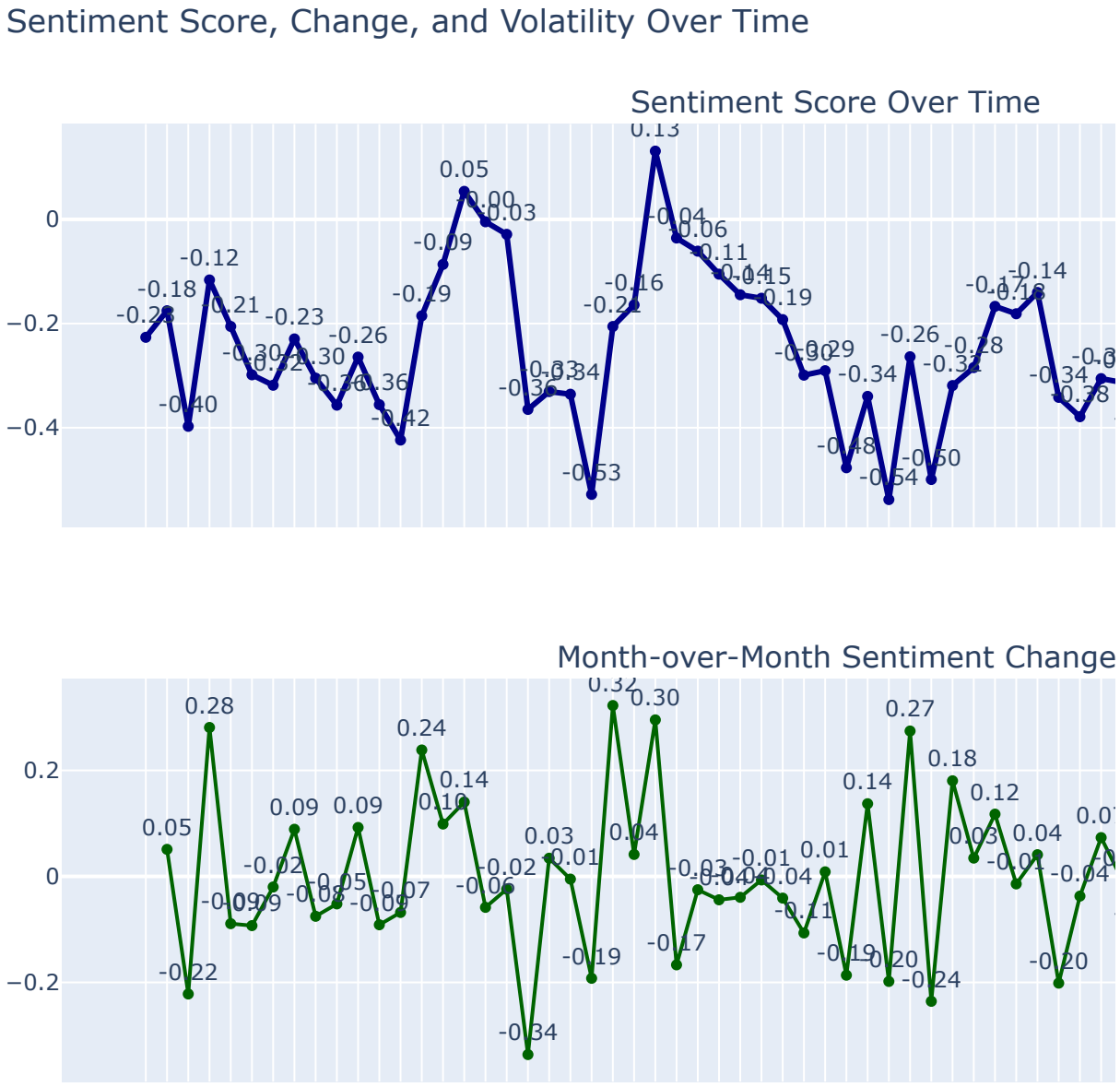


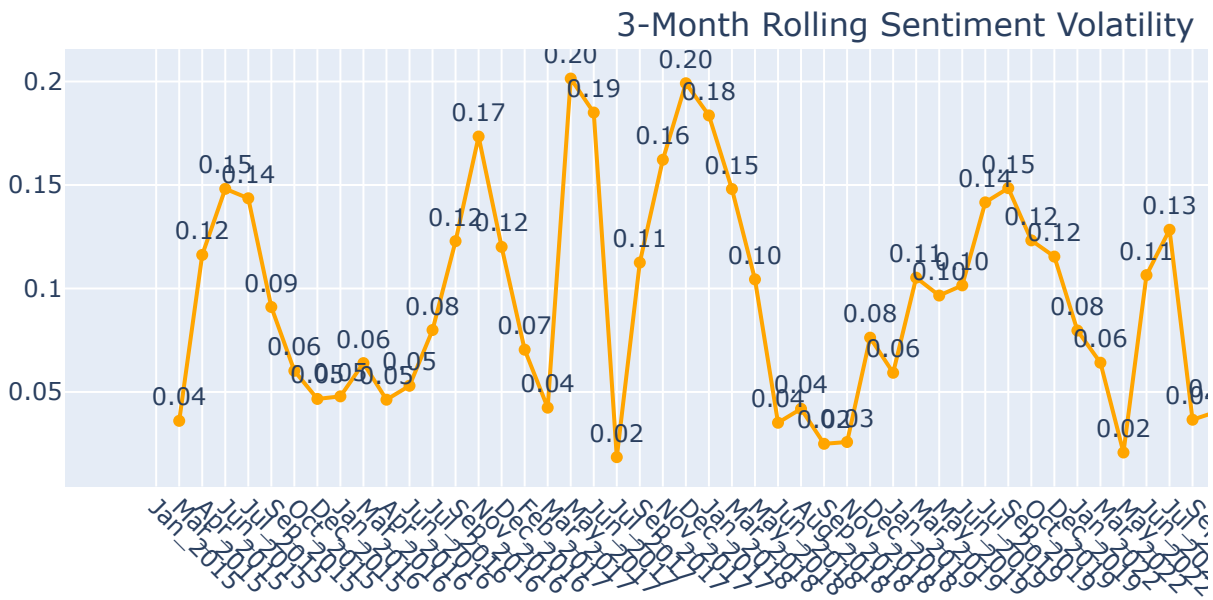

```
import pandas as pd
import plotly.graph_objects as go
from plotly.subplots import make_subplots

# Sort df_1 and format date
df_1 = df_1.sort_values(by='Date')
df_1['date_new'] = df_1['Date'].dt.strftime('%b_%Y')

# Compute all required metrics
df_1['sentiment_delta'] = df_1['sentiment_score'].diff()
df_1['sentiment_volatility'] = df_1['sentiment_score'].rolling(window=3, min_periods=1).s
df_1['cumulative_sentiment'] = df_1['sentiment_score'].cumsum()

# ------------------- PLOTS 1–3: Main Trend Metrics -------------------
fig = make_subplots(
    rows=3, cols=1,
    shared_xaxes=True,
    vertical_spacing=0.1,
    subplot_titles=[
        "Sentiment Score Over Time",
        "Month-over-Month Sentiment Change",
        "3-Month Rolling Sentiment Volatility"
    ]
)

# Plot 1: Sentiment Score
fig.add_trace(go.Scatter(
    x=df_1['date_new'],
    y=df_1['sentiment_score'],
```

```
    mode='lines+markers+text',
    text=[f"{v:.2f}" for v in df_1['sentiment_score']],
    textposition="top center",
    name="Sentiment Score",
    line=dict(color='darkblue', width=3)
), row=1, col=1)

# Plot 2: MoM Change
fig.add_trace(go.Scatter(
    x=df_1['date_new'],
    y=df_1['sentiment_delta'],
    mode='lines+markers+text',
    text=[f"{v:.2f}" if pd.notna(v) else "" for v in df_1['sentiment_delta']],
    textposition="top center",
    name="Sentiment Change",
    line=dict(color='darkgreen')
), row=2, col=1)

# Plot 3: Volatility
fig.add_trace(go.Scatter(
    x=df_1['date_new'],
    y=df_1['sentiment_volatility'],
    mode='lines+markers+text',
    text=[f"{v:.2f}" if pd.notna(v) else "" for v in df_1['sentiment_volatility']],
    textposition="top center",
    name="Volatility (3-Month Std Dev)",
    line=dict(color='orange')
), row=3, col=1)

fig.update_layout(
    height=1000,
    width=1000,
    title="Sentiment Score, Change, and Volatility Over Time",
    showlegend=False
)

fig.update_xaxes(tickangle=45, tickmode='array', tickvals=df_1['date_new'])
fig.show()

# ------------------- PLOT 4: Cumulative Sentiment -------------------
fig_cumulative = go.Figure()
fig_cumulative.add_trace(go.Scatter(
    x=df_1['date_new'],
    y=df_1['cumulative_sentiment'],
    mode='lines+markers+text',
    text=[f"{v:.2f}" for v in df_1['cumulative_sentiment']],
    textposition="top center",
    name='Cumulative Sentiment',
    line=dict(color='indigo')
))
fig_cumulative.update_layout(
```

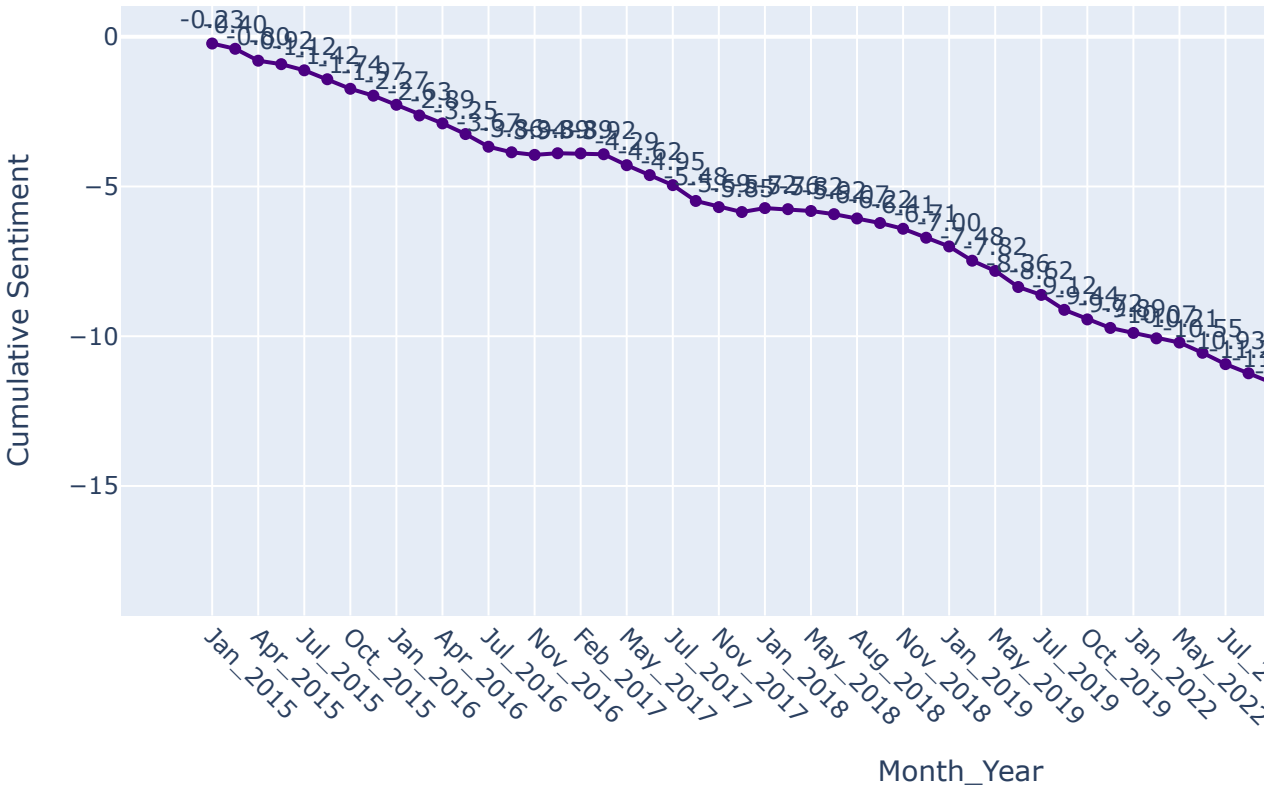
```
    title='Cumulative Sentiment Score Over Time',
    xaxis_title='Month_Year',
    yaxis_title='Cumulative Sentiment',
    xaxis=dict(tickangle=45),
    height=500,
    width=1000
)
fig_cumulative.show()

# ------------------- PLOT 5: Histogram of Sentiment Scores -------------------
fig_hist = go.Figure()
fig_hist.add_trace(go.Histogram(
    x=df_1['sentiment_score'],
    nbinsx=10,
    name='Sentiment Score Distribution',
    marker=dict(color='firebrick')
))
fig_hist.update_layout(
    title="Distribution of Sentiment Scores",
    xaxis_title="Sentiment Score",
    yaxis_title="Frequency",
    height=400,
    width=800
)
fig_hist.show()

# ------------------- PLOT 6: Annotated Min and Max Sentiment -------------------
fig_min_max = go.Figure()
fig_min_max.add_trace(go.Scatter(
    x=df_1['date_new'],
    y=df_1['sentiment_score'],
    mode='lines+markers',
    name='Sentiment Score',
    line=dict(color='slateblue')
))

min_idx = df_1['sentiment_score'].idxmin()
max_idx = df_1['sentiment_score'].idxmax()

fig_min_max.add_trace(go.Scatter(
    x=[df_1.loc[min_idx, 'date_new']],
    y=[df_1.loc[min_idx, 'sentiment_score']],
    text=["Min"],
    mode="markers+text",
    marker=dict(color="red", size=10),
    textposition="bottom center",
    name="Min Score"
))
fig_min_max.add_trace(go.Scatter(
    x=[df_1.loc[max_idx, 'date_new']],
```

```
    y=[df_1.loc[max_idx, 'sentiment_score']],
    text=["Max"],
    mode="markers+text",
    marker=dict(color="green", size=10),
    textposition="top center",
    name="Max Score"
))
fig_min_max.update_layout(
    title="Sentiment Score with Min/Max Annotations",
    xaxis_title="Month_Year",
    yaxis_title="Sentiment Score",
    xaxis=dict(tickangle=45),
    height=500,
    width=1000
)
fig_min_max.show()
```

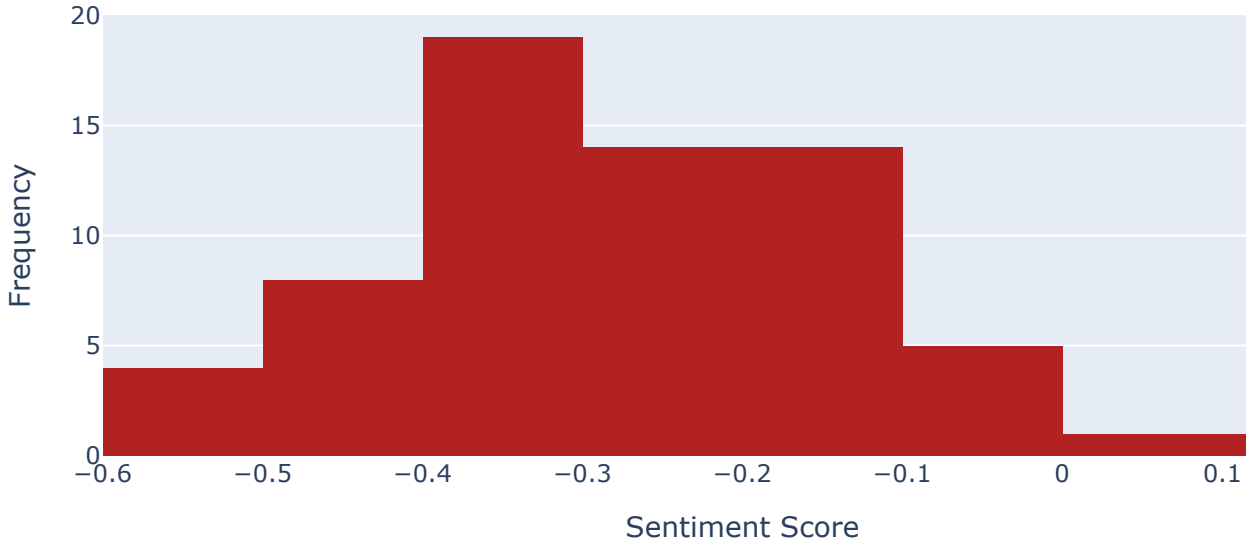


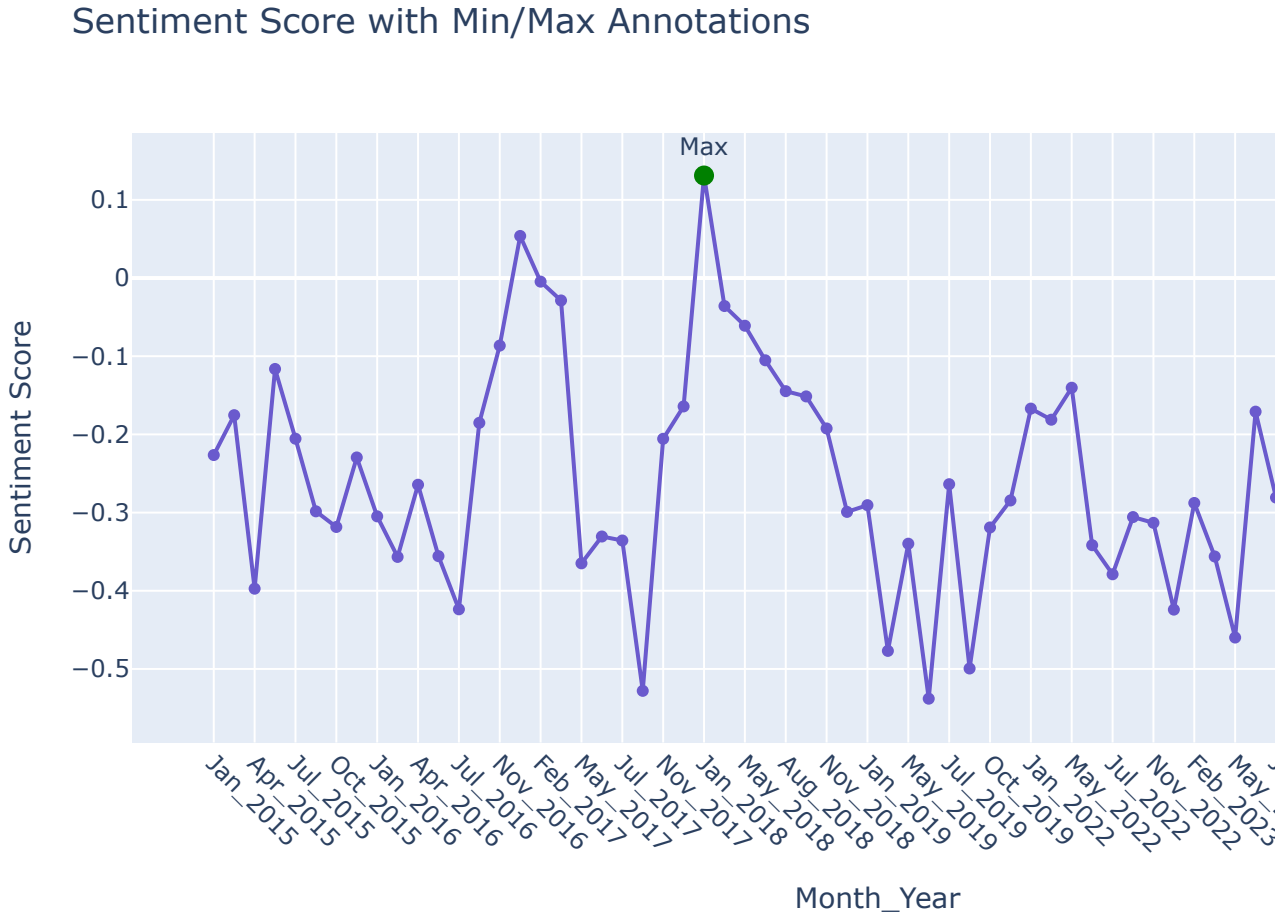


Cumulative Sentiment Score Over Time



Distribution of Sentiment Scores





Generate WordClouds (Yearly)

```
from wordcloud import WordCloud

years = sorted(df['Year'].unique())
```

```
ncols = 3
nrows = (len(years) + ncols - 1) // ncols

plt.figure(figsize=(20, 5 * nrows))

for i, year in enumerate(years):
    year_text = ' '.join(df[df['Year'] == year]['cleaned_text'].astype(str))
    wordcloud = WordCloud(width=800, height=400, background_color='white').generate(year_

    plt.subplot(nrows, ncols, i + 1)
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')
    plt.title(f'WordCloud - {year}')

plt.tight_layout()
plt.show()
```