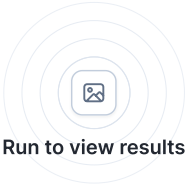
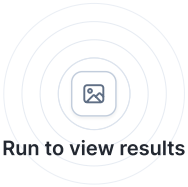
```
import nltk
nltk.download('vader_lexicon')
```



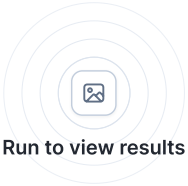

```
import pandas as pd
from gensim.models import Doc2Vec
from gensim.models.doc2vec import TaggedDocument
import gensim.downloader as api
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from sentence_transformers import SentenceTransformer, util # SBERT for sentence embeddings
from transformers import pipeline
import numpy as np
```

```
# Load the labeled Tweets dataset
df = pd.read_csv("/work/labeled_tweets.csv")
tweets = df['Tweet'].values
```



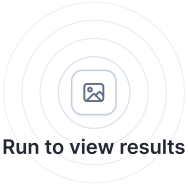

```
# Load pre-trained SBERT model
model = SentenceTransformer('paraphrase-MiniLM-L6-v2')
```

```python
# Generate embeddings
tweet_embeddings = model.encode(tweets)

# Ensure that the number of rows in tweets matches the number of embeddings
assert len(tweets) == tweet_embeddings.shape[0], "Number of tweets and embeddings do not match!"

# Convert embeddings array to DataFrame with appropriate column names
embedding_columns = [f"embedding_{i}" for i in range(tweet_embeddings.shape[1])]
embeddings_df = pd.DataFrame(tweet_embeddings, columns=embedding_columns)

# Concatenate the tweets and embeddings DataFrames
tweets_with_embeddings = pd.concat([df, embeddings_df], axis=1)

# Save the combined DataFrame so you do not have to wait 15 minutes to embed each time you run this notebook
tweets_with_embeddings.to_csv('embedded_tweets.csv', index=False)

# Display the combined DataFrame
tweets_with_embeddings
```



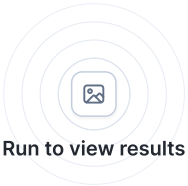
Run to view results

```python
# You can start from here if you want
df = pd.read_csv("embedded_tweets.csv")

# Select all embedding columns
embedding_columns = [col for col in df.columns if col.startswith('embedding_')]

# Convert the embedding columns to a NumPy array
tweet_embeddings = df[embedding_columns].to_numpy()

# Display the shape to confirm
print("Shape of embeddings array:", tweet_embeddings.shape)
df
```



Run to view results

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

# Create an empty data frame for the sentiment results
sentiment_tweets = df
sentiment_tweets.insert(3, 'Sentiment_Scores', np.nan)
sentiment_tweets.insert(4, 'Negativity', np.nan)
sentiment_tweets.insert(5, 'Neutrality', np.nan)
sentiment_tweets.insert(6, 'Positivity', np.nan)
sentiment_tweets.insert(7, 'Compound', np.nan)
sentiment_tweets['Tweet'] = sentiment_tweets['Tweet'].fillna("").astype(str)

# Run VADER analysis. Save the results in a file
sentiment_tweets['Sentiment_Scores'] = sentiment_tweets['Tweet'].apply(vader_analyzer.polarity_scores)
sentiment_tweets[['Negativity', 'Neutrality', 'Positivity', 'Compound']] = sentiment_tweets['Sentiment_Scores'].apply(lam
sentiment_tweets.to_csv('sentiment_tweets.csv', index=False)
sentiment_tweets
```



Run to view results