```
!pip install vaderSentiment
```

```
Collecting vaderSentiment
  Downloading vaderSentiment-3.3.2-py2.py3-none-any.whl.metadata (572 bytes)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from vaderSentiment) (2.32.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->vaderSentiment)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->vaderSentiment) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->vaderSentiment) (2.3.0
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->vaderSentiment) (2025.1
Downloading vaderSentiment-3.3.2-py2.py3-none-any.whl (125 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 126.0/126.0 kB 8.7 MB/s eta 0:00:00
Installing collected packages: vaderSentiment
Successfully installed vaderSentiment-3.3.2
```

```
# Core libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# NLP & ML
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
import nltk
from textblob import TextBlob
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

# Toxicity & source
import requests
# Core libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt



true_df = pd.read_csv('/content/True.csv')
fake_df = pd.read_csv('/content/Fake.csv')

true_df['label'] = 1
fake_df['label'] = 0

df = pd.concat([true_df, fake_df]).sample(frac=1).reset_index(drop=True)
```

For fake news: high precision avoids false accusations.

For spam detection: high recall avoids missing any spam.

1. TRUTH AND ACCURACY

```
vectorizer = TfidfVectorizer(stop_words='english', max_df=0.7)
X = vectorizer.fit_transform(df['text'])
y = df['label']

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

model = LogisticRegression()
model.fit(X_train, y_train)

pred = model.predict(X_test)
print(classification_report(y_test, pred))
```

```
              precision    recall  f1-score   support

           0       0.99      0.98      0.99      4658
           1       0.98      0.99      0.99      4322

    accuracy                           0.99      8980
   macro avg       0.99      0.99      0.99      8980
weighted avg       0.99      0.99      0.99      8980
```

2. INDEPENDENCE- SOURCE BIAS CHECKER

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
import pandas as pd

df = pd.read_excel('/content/annotations.xlsx')
df = df[['outlet', 'label', 'text', 'link', 'political_ideology']]
df.head()
```

	outlet	label	text	link	political_ideology
0	breitbart	Non-biased	The transgender effort to suppress any recogni...	https://www.breitbart.com/politics/2019/02/21/...	7
1	alternet	Non-biased	Radical Virginia Citizens Defense League has o...	https://www.alternet.org/2020/01/pro-gun-prote...	7
2	msnbc	Non-biased	Miller is the architect of President Donald Tr...	https://www.nbcnews.com/news/latino/after-step...	7
3	breitbart	Non-biased	The House Democrats' 1,400-page coronavirus re...	https://www.breitbart.com/politics/2020/03/24/...	7
4	federalist	Non-biased	A specter is haunting the West; our elites see...	https://thefederalist.com/2019/11/08/nationali...	7

```
df['label'].value_counts()
```

label	count
Biased	10651
Non-biased	7124

dtype: int64

```
bias_map = {
    'left': -1,
    'center': 0,
    'right': 1,
    'biased': 1,
    'neutral': 0,
    'unbiased': 0,
    # adjust based on actual values in `label`
}
df['bias_score'] = df['label'].map(bias_map)
```

new one

```
!pip install scrapy beautifulsoup4 pandas anonymizer
```

```
Collecting scrapy
  Using cached Scrapy-2.12.0-py2.py3-none-any.whl.metadata (5.3 kB)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.11/dist-packages (4.13.3)
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)
Collecting anonymizer
  Downloading anonymizer-0.0.6.tar.gz (7.1 kB)
  Preparing metadata (setup.py) ... done
Collecting Twisted>=21.7.0 (from scrapy)
  Downloading twisted-24.11.0-py3-none-any.whl.metadata (20 kB)
Requirement already satisfied: cryptography>=37.0.0 in /usr/local/lib/python3.11/dist-packages (from scrapy) (43.0.3)
Collecting cssselect>=0.9.1 (from scrapy)
  Downloading cssselect-1.3.0-py3-none-any.whl.metadata (2.6 kB)
Collecting itemloaders>=1.0.1 (from scrapy)
  Downloading itemloaders-1.3.2-py3-none-any.whl.metadata (3.9 kB)
Collecting parsel>=1.5.0 (from scrapy)
  Downloading parsel-1.10.0-py2.py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: pyOpenSSL>=22.0.0 in /usr/local/lib/python3.11/dist-packages (from scrapy) (24.2.1)
Collecting queuelib>=1.4.2 (from scrapy)
  Downloading queuelib-1.8.0-py3-none-any.whl.metadata (6.1 kB)
Collecting service-identity>=18.1.0 (from scrapy)
  Downloading service_identity-24.2.0-py3-none-any.whl.metadata (5.1 kB)
Collecting w3lib>=1.17.0 (from scrapy)
  Downloading w3lib-2.3.1-py3-none-any.whl.metadata (2.3 kB)
Collecting zope.interface>=5.1.0 (from scrapy)
  Downloading zope.interface-7.2-cp311-cp311-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.wh
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.4/44.4 kB 3.1 MB/s eta 0:00:00
Collecting protego>=0.1.15 (from scrapy)
```

```
  Downloading Protego-0.4.0-py2.py3-none-any.whl.metadata (6.2 kB)
Collecting itemadapter>=0.1.0 (from scrapy)
  Downloading itemadapter-0.11.0-py3-none-any.whl.metadata (18 kB)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from scrapy) (24.2)
Collecting tldextract (from scrapy)
  Downloading tldextract-5.1.3-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: lxml>=4.6.0 in /usr/local/lib/python3.11/dist-packages (from scrapy) (5.3.1)
Requirement already satisfied: defusedxml>=0.7.1 in /usr/local/lib/python3.11/dist-packages (from scrapy) (0.7.1)
Collecting PyDispatcher>=2.0.5 (from scrapy)
  Downloading PyDispatcher-2.0.7-py3-none-any.whl.metadata (2.4 kB)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.11/dist-packages (from beautifulsoup4) (2.6)
Requirement already satisfied: typing-extensions>=4.0.0 in /usr/local/lib/python3.11/dist-packages (from beautifulsoup4) (4.13.0)
Requirement already satisfied: numpy>=1.23.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: cffi>=1.12 in /usr/local/lib/python3.11/dist-packages (from cryptography>=37.0.0->scrapy) (1.17.1)
Collecting jmespath>=0.9.5 (from itemloaders>=1.0.1->scrapy)
  Downloading jmespath-1.0.1-py3-none-any.whl.metadata (7.6 kB)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Requirement already satisfied: attrs>=19.1.0 in /usr/local/lib/python3.11/dist-packages (from service-identity>=18.1.0->scrapy) (
Requirement already satisfied: pyasn1 in /usr/local/lib/python3.11/dist-packages (from service-identity>=18.1.0->scrapy) (0.6.1)
Requirement already satisfied: pyasn1-modules in /usr/local/lib/python3.11/dist-packages (from service-identity>=18.1.0->scrapy)
Collecting automat>=24.8.0 (from Twisted>=21.7.0->scrapy)
  Downloading Automat-24.8.1-py3-none-any.whl.metadata (8.4 kB)
Collecting constantly>=15.1 (from Twisted>=21.7.0->scrapy)
  Downloading constantly-23.10.4-py3-none-any.whl.metadata (1.8 kB)
Collecting hyperlink>=17.1.1 (from Twisted>=21.7.0->scrapy)
  Downloading hyperlink-21.0.0-py2.py3-none-any.whl.metadata (1.5 kB)
Collecting incremental>=24.7.0 (from Twisted>=21.7.0->scrapy)
```

to return ethically scraped public articles

```
import requests
from bs4 import BeautifulSoup
import re

def ethical_scraper(url):
    response = requests.get(url)
    if response.status_code == 200:
        soup = BeautifulSoup(response.text, 'html.parser')
        # Exclude private content
        private_content = soup.find_all(class_=re.compile('private|restricted'))
        public_content = [c for c in soup.find_all('article') if c not in private_content]
        return public_content
```

```
!pip install anonymizer
```

```
Requirement already satisfied: anonymizer in /usr/local/lib/python3.11/dist-packages (0.0.6)
```

Double-click (or enter) to edit

```
# K-anonymity implementation
   from anonymizer import Anonymizer
   import pandas as pd

   def anonymize_data(df, sensitive_columns):
       anonymizer = Anonymizer(df)
       return anonymizer.k_anonymize(k=3, columns=sensitive_columns)

   # Example usage
   df = pd.read_csv('journalism_data.csv')
   safe_df = anonymize_data(df, ['location', 'age'])
```

```
import nltk
nltk.download('punkt_tab') # Download the 'punkt_tab' data
from scipy.stats import chisquare
import pandas as pd

# Download required NLTK resources
nltk.download('opinion_lexicon')
nltk.download('punkt')

from nltk.corpus import opinion_lexicon
from nltk.tokenize import word_tokenize
from nltk.probability import FreqDist

def calculate_bias(text):
```

```
    pos_words = set(opinion_lexicon.positive())
    neg_words = set(opinion_lexicon.negative())

    tokens = word_tokenize(text.lower())  # Lowercase for matching
    word_counts = FreqDist(tokens)

    pos_count = sum(word_counts[w] for w in pos_words if w in word_counts)
    neg_count = sum(word_counts[w] for w in neg_words if w in word_counts)

    if pos_count + neg_count == 0:
        return {"pos": 0, "neg": 0, "bias": "No opinion words", "p_value": None}

    result = chisquare([pos_count, neg_count])
    return {
        "pos": pos_count,
        "neg": neg_count,
        "bias": "Positive" if pos_count > neg_count else "Negative" if neg_count > pos_count else "Neutral",
        "p_value": result.pvalue
    }

# Example: Test it on a sample text
sample = "The economy is amazing, but the leadership is corrupt and unfair."
print(calculate_bias(sample))
```

```
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
[nltk_data] Downloading package opinion_lexicon to /root/nltk_data...
[nltk_data]   Package opinion_lexicon is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
{'pos': 1, 'neg': 1, 'bias': 'Neutral', 'p_value': np.float64(1.0)}
```

```
import pandas as pd

# Try specifying a different delimiter, or using the 'error_bad_lines' argument
df = pd.read_csv('/content/bbc-news-data.csv', sep='\t')  # or sep=',', etc.
# Alternatively, if you want to keep the bad lines, use 'warn' instead of 'skip'
#df = pd.read_csv('/content/bbc-news-data.csv', sep=',', on_bad_lines='warn')  # or sep='\t', etc.

df.head()
```

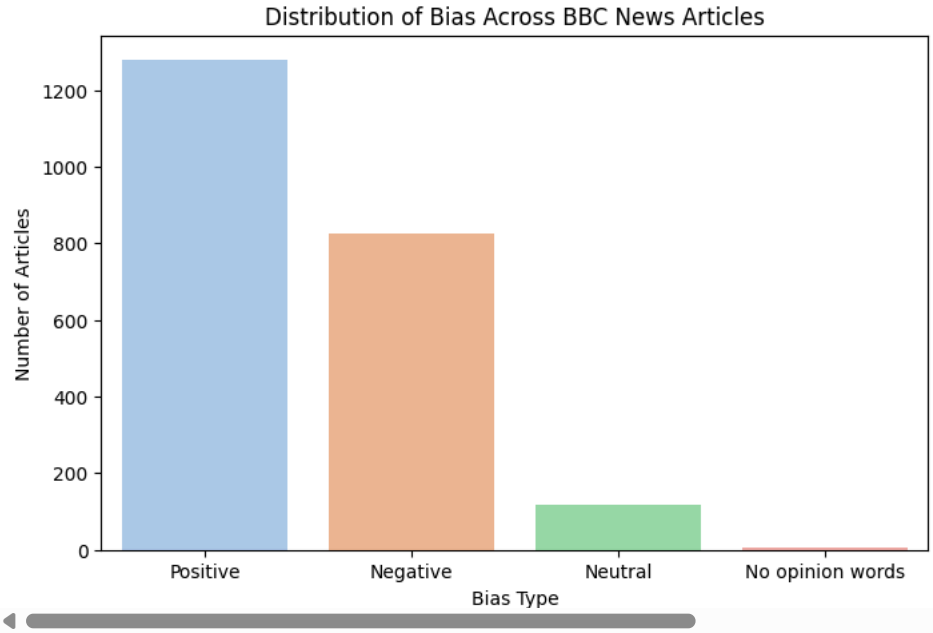
	category	filename	title	content
0	business	001.txt	Ad sales boost Time Warner profit	Quarterly profits at US media giant TimeWarne...
1	business	002.txt	Dollar gains on Greenspan speech	The dollar has hit its highest level against ...
2	business	003.txt	Yukos unit buyer faces loan claim	The owners of embattled Russian oil giant Yuk...
3	business	004.txt	High fuel prices hit BA's profits	British Airways has blamed high fuel prices f...
4	business	005.txt	Pernod takeover talk lifts Domecq	Shares in UK drinks and food firm Allied Dome...

```
print(df.columns)
```

```
Index(['category', 'filename', 'title', 'content'], dtype='object')
```

```
df['bias_result'] = df['content'].apply(calculate_bias)
df['bias_label'] = df['bias_result'].apply(lambda x: x['bias'])



import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 5))
sns.countplot(x='bias_label', data=df, palette='pastel')
plt.title('Distribution of Bias Across BBC News Articles')
plt.xlabel('Bias Type')
plt.ylabel('Number of Articles')
plt.show()
```

```
<ipython-input-58-fe34783eeaf7>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

  sns.countplot(x='bias_label', data=df, palette='pastel')
```




```
!pip install lifelines
```

```
Collecting lifelines
  Downloading lifelines-0.30.0-py3-none-any.whl.metadata (3.2 kB)
Requirement already satisfied: numpy>=1.14.0 in /usr/local/lib/python3.11/dist-packages (from lifelines) (2.0.2)
Requirement already satisfied: scipy>=1.7.0 in /usr/local/lib/python3.11/dist-packages (from lifelines) (1.14.1)
Requirement already satisfied: pandas>=2.1 in /usr/local/lib/python3.11/dist-packages (from lifelines) (2.2.2)
Requirement already satisfied: matplotlib>=3.0 in /usr/local/lib/python3.11/dist-packages (from lifelines) (3.10.0)
Requirement already satisfied: autograd>=1.5 in /usr/local/lib/python3.11/dist-packages (from lifelines) (1.7.0)
Collecting autograd-gamma>=0.3 (from lifelines)
  Downloading autograd-gamma-0.5.0.tar.gz (4.0 kB)
  Preparing metadata (setup.py) ... done
Collecting formulaic>=0.2.2 (from lifelines)
  Downloading formulaic-1.1.1-py3-none-any.whl.metadata (6.9 kB)
Collecting interface-meta>=1.2.0 (from formulaic>=0.2.2->lifelines)
  Downloading interface_meta-1.3.0-py3-none-any.whl.metadata (6.7 kB)
Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/lib/python3.11/dist-packages (from formulaic>=0.2.2->lifelines
Requirement already satisfied: wrapt>=1.0 in /usr/local/lib/python3.11/dist-packages (from formulaic>=0.2.2->lifelines) (1.17.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=3.0->lifelines) (1.3.1
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=3.0->lifelines) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=3.0->lifelines) (4.56
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=3.0->lifelines) (1.4.8
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=3.0->lifelines) (24.2)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=3.0->lifelines) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=3.0->lifelines) (3.2.3
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.11/dist-packages (from matplotlib>=3.0->lifelines) (2
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas>=2.1->lifelines) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas>=2.1->lifelines) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.7->matplotlib>=3.0->life
Downloading lifelines-0.30.0-py3-none-any.whl (349 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 349.3/349.3 kB 19.8 MB/s eta 0:00:00
Downloading formulaic-1.1.1-py3-none-any.whl (115 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 115.7/115.7 kB 9.2 MB/s eta 0:00:00
Downloading interface_meta-1.3.0-py3-none-any.whl (14 kB)
Building wheels for collected packages: autograd-gamma
  Building wheel for autograd-gamma (setup.py) ... done
  Created wheel for autograd-gamma: filename=autograd_gamma-0.5.0-py3-none-any.whl size=4030 sha256=fc5e5094769ceea12006d9035eddb4c7
  Stored in directory: /root/.cache/pip/wheels/8b/67/f4/2caaae2146198dcb824f31a303833b07b14a5ec863fb3acd7b
Successfully built autograd-gamma
Installing collected packages: interface-meta, autograd-gamma, formulaic, lifelines
Successfully installed autograd-gamma-0.5.0 formulaic-1.1.1 interface-meta-1.3.0 lifelines-0.30.0
```

churn means:

A reader stops visiting the news site

A subscriber cancels their membership

A viewer stops interacting with news on social media

```
import pandas as pd
import numpy as np
from lifelines import KaplanMeierFitter
import matplotlib.pyplot as plt

# Simulate dummy 'engagement duration' and 'churn' status
np.random.seed(42)  # for reproducibility
df['days_subscribed'] = np.random.randint(1, 100, size=len(df))  # how many days it was read or relevant
df['churned'] = np.random.choice([1, 0], size=len(df), p=[0.7, 0.3])  # 70% churned, 30% still engaging
```

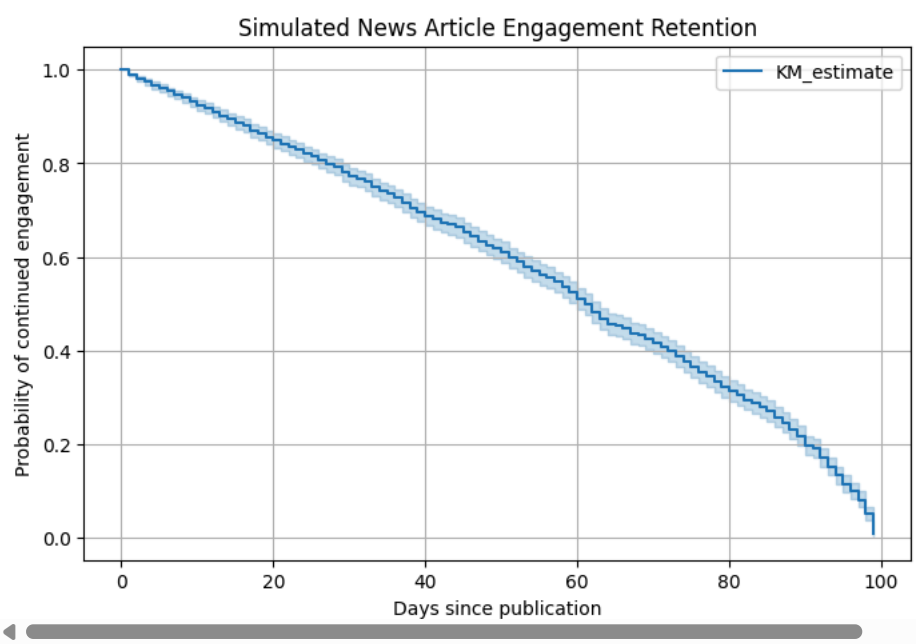
a Kaplan-Meier survival curve of user retention.

The curve shows how long articles tend to stay engaging.

A steeper drop means quick drop-off in interest.

Flattening tail means some articles maintain long-term value.

```
# Initialize Kaplan-Meier model
kmf = KaplanMeierFitter()

# Fit model
kmf.fit(durations=df['days_subscribed'], event_observed=df['churned'])

# Plot survival (retention) function
plt.figure(figsize=(8, 5))
kmf.plot_survival_function()
plt.title('Simulated News Article Engagement Retention')
plt.xlabel('Days since publication')
plt.ylabel('Probability of continued engagement')
plt.grid(True)
plt.savefig('churn_analysis.png')
plt.show()
```




```
!pip install Jinja2
```

```
Requirement already satisfied: Jinja2 in /usr/local/lib/python3.11/dist-packages (3.1.6)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-packages (from Jinja2) (3.0.2)
```

```
import pandas as pd
from jinja2 import Template
from datetime import datetime
import random

# Simulate 'location' and 'date' columns
locations = ['London', 'New York', 'Delhi', 'Paris', 'Berlin', 'Toronto']
df['location'] = [random.choice(locations) for _ in range(len(df))]
df['date'] = datetime.today().strftime('%Y-%m-%d')
df['summary'] = df['content'].str.slice(0, 150) + '...'  # First 150 chars as summary

# Template for report
template = Template("""
```

```
🗞️ Breaking News: {{ headline }}
📍 Location: {{ location }}
📅 Date: {{ date }}
📝 Summary: {{ summary }}
""")

# Function to generate one news report
def generate_news_report(data):
    return template.render(**data)

# Apply it to each row
df['report'] = df.apply(lambda row: generate_news_report({
    'headline': row['title'],
    'location': row['location'],
    'date': row['date'],
    'summary': row['summary']
}), axis=1)

# Show a few sample reports
for i, report in enumerate(df['report'].head(3), 1):
    print(f"\n--- News Report {i} ---\n{report}")
```

```
--- News Report 1 ---

🗞️ Breaking News: Ad sales boost Time Warner profit
📍 Location: Berlin
📅 Date: 2025-04-05
📝 Summary:  Quarterly profits at US media giant TimeWarner jumped 76% to $1.13bn (£600m) for the three months to December, from $6

--- News Report 2 ---

🗞️ Breaking News: Dollar gains on Greenspan speech
📍 Location: Toronto
📅 Date: 2025-04-05
📝 Summary:  The dollar has hit its highest level against the euro in almost three months after the Federal Reserve head said the U

--- News Report 3 ---

🗞️ Breaking News: Yukos unit buyer faces loan claim
📍 Location: Delhi
📅 Date: 2025-04-05
📝 Summary:  The owners of embattled Russian oil giant Yukos are to ask the buyer of its former production unit to pay back a $900m
```

```
!pip install dash plotly nltk pandas
```

```
Requirement already satisfied: dash in /usr/local/lib/python3.11/dist-packages (3.0.2)
Requirement already satisfied: plotly in /usr/local/lib/python3.11/dist-packages (5.24.1)
Requirement already satisfied: nltk in /usr/local/lib/python3.11/dist-packages (3.9.1)
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)
Requirement already satisfied: Flask<3.1,>=1.0.4 in /usr/local/lib/python3.11/dist-packages (from dash) (3.0.3)
Requirement already satisfied: Werkzeug<3.1 in /usr/local/lib/python3.11/dist-packages (from dash) (3.0.6)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.11/dist-packages (from dash) (8.6.1)
Requirement already satisfied: typing-extensions>=4.1.1 in /usr/local/lib/python3.11/dist-packages (from dash) (4.13.0)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from dash) (2.32.3)
Requirement already satisfied: retrying in /usr/local/lib/python3.11/dist-packages (from dash) (1.3.4)
Requirement already satisfied: nest-asyncio in /usr/local/lib/python3.11/dist-packages (from dash) (1.6.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/dist-packages (from dash) (75.2.0)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.11/dist-packages (from plotly) (9.1.2)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from plotly) (24.2)
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packages (from nltk) (8.1.8)
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-packages (from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.11/dist-packages (from nltk) (2024.11.6)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from nltk) (4.67.1)
Requirement already satisfied: numpy>=1.23.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: Jinja2>=3.1.2 in /usr/local/lib/python3.11/dist-packages (from Flask<3.1,>=1.0.4->dash) (3.1.6)
Requirement already satisfied: itsdangerous>=2.1.2 in /usr/local/lib/python3.11/dist-packages (from Flask<3.1,>=1.0.4->dash) (2.2.0
Requirement already satisfied: blinker>=1.6.2 in /usr/local/lib/python3.11/dist-packages (from Flask<3.1,>=1.0.4->dash) (1.9.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.11/dist-packages (from Werkzeug<3.1->dash) (3.0.2)
Requirement already satisfied: zipp>=3.20 in /usr/local/lib/python3.11/dist-packages (from importlib-metadata->dash) (3.21.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->dash) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->dash) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->dash) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->dash) (2025.1.31)
```

```
import nltk
from nltk.corpus import opinion_lexicon
```

```
from nltk.tokenize import word_tokenize
nltk.download('opinion_lexicon')
nltk.download('punkt')

def calculate_bias_score(text):
    pos_words = set(opinion_lexicon.positive())
    neg_words = set(opinion_lexicon.negative())
    words = word_tokenize(str(text).lower())

    pos_count = sum(1 for w in words if w in pos_words)
    neg_count = sum(1 for w in words if w in neg_words)
    total = pos_count + neg_count

    if total == 0:
        return 0  # Neutral
    return (pos_count - neg_count) / total  # Positive = bias toward positivity, Negative = negativity
```

```
[nltk_data] Downloading package opinion_lexicon to /root/nltk_data...
[nltk_data]   Package opinion_lexicon is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

```
import pandas as pd

# For demo: Add fake dates (or use your own if available)
df['date'] = pd.date_range(end=pd.Timestamp.today(), periods=len(df))

# Compute bias score
df['bias_score'] = df['content'].apply(calculate_bias_score)

# Group by date to get average bias
trend_df = df.groupby(df['date'].dt.date)['bias_score'].mean().reset_index()
trend_df.columns = ['Date', 'Bias Score']
```

```
import dash
from dash import dcc, html
from dash.dependencies import Input, Output
import plotly.express as px

# Initialize app
app = dash.Dash(__name__)
app.title = "BBC Bias Trend"

# Layout
app.layout = html.Div([
    html.H2("BBC News Bias Trend"),
    dcc.Graph(id='bias-trend'),
    dcc.Interval(id='interval', interval=86400000, n_intervals=0),  # Daily refresh
])

# Callback to update graph
@app.callback(
    Output('bias-trend', 'figure'),
    Input('interval', 'n_intervals')
)
def update_graph(n):
    fig = px.line(trend_df, x='Date', y='Bias Score', title='Bias Score Over Time')
    fig.update_layout(yaxis_title='Bias Score (pos-neg)', xaxis_title='Date')
    return fig

# Run server
app.run(debug=False, port=8050)
```