# ✅ STEP 0: Install required packages

!pip install requests feedparser tqdm chardet --quiet

import requests

import feedparser

import pandas as pd

from datetime import datetime

from tqdm import tqdm

from urllib.parse import quote

import socket

# --- Prevent feedparser from hanging ---

socket.setdefaulttimeout(20)

# ✅ STEP 1: Stock/company-specific news sources

NEWSDATA\_API\_KEY = "pub\_55b70ce890df4aec9ddd99936d122e86" # <-- YOUR KEY

symbols = [

"TATA MOTORS", "RELIANCE", "INFOSYS", "HDFC BANK", "ICICI BANK",

"ITC", "SBIN", "WIPRO", "BHARTI AIRTEL", "ADANI"

]

def fetch\_newsdata\_articles(stock\_name, max\_articles=10):

url = f"https://newsdata.io/api/1/news?apikey={NEWSDATA\_API\_KEY}&q={quote(stock\_name)}&country=in&language=en&category=business,top,world"

all\_articles = []

try:

r = requests.get(url, timeout=15)

if r.status\_code == 200:

resp = r.json()

for article in resp.get("results", [])[:max\_articles]:

all\_articles.append({

"stock": stock\_name,

"title": article.get("title"),

"description": article.get("description"),

"pubDate": article.get("pubDate"),

"link": article.get("link"),

"source": article.get("source\_id", "NewsData.io")

})

except Exception as e:

print(f"❌ Error NewsData {stock\_name}: {e}")

return all\_articles

newsdata\_news = []

for stock in tqdm(symbols, desc="NewsData.io"):

newsdata\_news.extend(fetch\_newsdata\_articles(stock, max\_articles=5))

newsdata\_df = pd.DataFrame(newsdata\_news)

def fetch\_google\_news(stock\_name, max\_articles=10):

query = quote(f"{stock\_name} stock OR {stock\_name} NSE OR {stock\_name} BSE")

url = f"https://news.google.com/rss/search?q={query}+when:7d&hl=en-IN&gl=IN&ceid=IN:en"

feed = feedparser.parse(url)

articles = []

for entry in feed.entries[:max\_articles]:

articles.append({

"stock": stock\_name,

"title": entry.get("title", ""),

"description": entry.get("description", ""),

"pubDate": entry.get("published", ""),

"link": entry.get("link", ""),

"source": entry.get("source", {}).get("title", "Google News") if entry.get("source") else "Google News"

})

return articles

google\_news = []

for stock in tqdm(symbols, desc="Google News"):

google\_news.extend(fetch\_google\_news(stock, max\_articles=5))

google\_news\_df = pd.DataFrame(google\_news)

def fetch\_yahoo\_news(stock\_code, max\_articles=10):

possible\_symbols = {

"TATA MOTORS": "TATAMOTORS.NS",

"RELIANCE": "RELIANCE.NS",

"INFOSYS": "INFY.NS",

"HDFC BANK": "HDFCBANK.NS",

"ICICI BANK": "ICICIBANK.NS",

"ITC": "ITC.NS",

"SBIN": "SBIN.NS",

"WIPRO": "WIPRO.NS",

"BHARTI AIRTEL": "BHARTIARTL.NS",

"ADANI": "ADANIENT.NS",

}

ticker = possible\_symbols.get(stock\_code.upper(), "")

if not ticker: return []

url = f"https://finance.yahoo.com/rss/headline?s={ticker}"

feed = feedparser.parse(url)

articles = []

for entry in feed.entries[:max\_articles]:

articles.append({

"stock": ticker,

"title": entry.get("title", ""),

"description": entry.get("description", ""),

"pubDate": entry.get("published", ""),

"link": entry.get("link", ""),

"source": "Yahoo Finance"

})

return articles

yahoo\_news = []

for stock in tqdm(symbols, desc="Yahoo Finance"):

yahoo\_news.extend(fetch\_yahoo\_news(stock, max\_articles=5))

yahoo\_news\_df = pd.DataFrame(yahoo\_news)

# ✅ STEP 2: Add ALL possible RSS sources (the giant list you gave!)

rss\_sources = [

# --- INDIAN NATIONAL BUSINESS/FINANCE ---

"https://economictimes.indiatimes.com/markets/rssfeeds/1977021501.cms",

"https://economictimes.indiatimes.com/news/economy/rssfeeds/1373380681.cms",

"https://economictimes.indiatimes.com/rssfeedsdefault.cms",

"https://www.moneycontrol.com/rss/markets.xml",

"https://www.moneycontrol.com/rss/buzzingstocks.xml",

"https://www.moneycontrol.com/rss/banking.xml",

"https://www.moneycontrol.com/rss/ipo.xml",

"https://www.moneycontrol.com/rss/commodities.xml",

"https://www.moneycontrol.com/rss/business.xml",

"https://www.moneycontrol.com/rss/economy.xml",

"https://www.business-standard.com/rss/markets-106.rss",

"https://www.business-standard.com/rss/companies-101.rss",

"https://www.business-standard.com/rss/economy-policy-102.rss",

"https://www.business-standard.com/rss/international-103.rss",

"https://www.livemint.com/rss/markets",

"https://www.livemint.com/rss/companies",

"https://www.livemint.com/rss/economy",

"https://www.livemint.com/rss/opinion",

"https://www.bqprime.com/feed",

"https://www.financialexpress.com/feed/",

"https://www.thehindubusinessline.com/markets/stock-markets/?service=rss",

"https://www.thehindubusinessline.com/economy/?service=rss",

"https://www.hindustantimes.com/business/rss/topnews/rssfeed.xml",

"https://www.indiatimes.com/business/rss",

"https://www.freepressjournal.in/business/feed",

"https://www.newindianexpress.com/Business/rssfeed/?id=196&getXmlFeed=true",

"https://zeenews.india.com/rss/business-news.xml",

"https://www.onmanorama.com/business/news.html/rssfeed.xml",

"https://www.deccanherald.com/business/rss-feeds",

"https://theprint.in/feed/",

# --- SECTORAL / THEMATIC ---

"https://energy.economictimes.indiatimes.com/rss/topstories.cms",

"https://realty.economictimes.indiatimes.com/rss/topstories.cms",

"https://auto.economictimes.indiatimes.com/rss/topstories.cms",

"https://retail.economictimes.indiatimes.com/rss/topstories.cms",

"https://cio.economictimes.indiatimes.com/rss/topstories.cms",

"https://bfsi.economictimes.indiatimes.com/rss/topstories.cms",

"https://health.economictimes.indiatimes.com/rss/topstories.cms",

"https://travel.economictimes.indiatimes.com/rss/topstories.cms",

"https://www.autocarindia.com/rss/news",

"https://www.cio.in/rss/news",

"https://www.pharmabiz.com/RSS/RssFeeds.aspx?cat=GENERAL",

"https://www.foodbusinessnews.net/rss/topic/97-fmcg",

"https://www.moneycontrol.com/rss/commodities.xml",

"https://economictimes.indiatimes.com/mf/rssfeeds/4693514.cms",

# --- IPOs, MUTUAL FUNDS ---

"https://www.moneycontrol.com/rss/ipo.xml",

"https://economictimes.indiatimes.com/markets/ipo/news/rssfeeds/52211617.cms",

"https://www.moneycontrol.com/rss/mf.xml",

"https://www.valueresearchonline.com/rss/mutual-funds/latest-news/",

# --- REGIONAL / INDIAN LANGUAGES ---

"https://www.bhaskar.com/rss-feed/2318/",

"https://www.jagran.com/rss/business.xml",

"https://navbharattimes.indiatimes.com/rssfeedstopstories.cms",

"https://www.divyabhaskar.co.in/rss-feed/25/",

"https://www.sandesh.com/rss/business.xml",

"https://www.dinamalar.com/rss\_section.asp?sec=business",

"https://www.manoramaonline.com/business/business-news.xml",

"https://www.mathrubhumi.com/business-news/rssfeed.xml",

"https://vijaykarnataka.com/rssfeeds/2280.cms",

"https://www.andhrajyothy.com/business/feed",

"https://www.loksatta.com/rss/section/business/",

"https://www.esakal.com/rss/section/business",

"https://www.anandabazar.com/business?service=rss",

"https://www.etemaaddaily.com/rss/biz.xml",

# --- GLOBAL BUSINESS & FINANCE ---

"http://feeds.reuters.com/reuters/businessNews",

"http://feeds.reuters.com/reuters/INbusinessNews",

"https://www.cnbc.com/id/10001147/device/rss/rss.html",

"https://feeds.a.dj.com/rss/RSSMarketsMain.xml",

"https://www.bloomberg.com/feed/podcast/bloomberg\_markets.xml",

"https://feeds.marketwatch.com/marketwatch/topstories/",

"https://www.nasdaq.com/feed/rssoutbound?category=Business",

"https://feeds.bbci.co.uk/news/business/rss.xml",

"https://www.ft.com/?format=rss",

"https://www.wsj.com/xml/rss/3\_7031.xml",

"https://www.forbes.com/business/feed/",

"https://www.seekingalpha.com/market\_currents.xml",

"https://www.investing.com/rss/news\_25.rss",

"https://asia.nikkei.com/rss/feed/nar",

"https://www.channelnewsasia.com/business/rss",

"https://www.businesstimes.com.sg/companies-markets/rss.xml",

"https://www.smh.com.au/rss/business.xml",

"https://www.theaustralian.com.au/business/latest-news/rss",

"https://www.afr.com/rss/feed/breaking-news-seo",

"https://english.etnews.com/rss/news.xml",

# --- MISC. (STARTUPS, TECH, POLICY, ETC.) ---

"https://inc42.com/feed/",

"https://yourstory.com/feed",

"https://www.vccircle.com/rssfeeds/allnews",

"https://www.entrackr.com/feed/",

"https://trak.in/tags/business/feed/",

"https://www.startupindia.gov.in/rss/news-feed.xml",

# --- CRYPTO/WEB3 ---

"https://cointelegraph.com/rss",

"https://news.bitcoin.com/feed/",

"https://www.coindesk.com/arc/outboundfeeds/rss/",

"https://www.newsbtc.com/feed/",

# --- US/EUROPE/ASIA/AFRICA REGIONAL BUSINESS ---

"https://feeds.skynews.com/feeds/rss/business.xml",

"https://www.africanews.com/feed/rss/business/",

"https://www.japantimes.co.jp/feed/",

"https://www.lemonde.fr/economie/rss\_full.xml",

"https://elpais.com/rss/america\_economia.xml",

# --- OTHER MAJOR NEWS SITES WITH BUSINESS SECTIONS ---

"https://www.aljazeera.com/xml/rss/all.xml",

"https://www.scmp.com/rss/91/feed",

"https://www.theguardian.com/uk/business/rss",

"https://rss.dw.com/rdf/rss-en-bus",

"https://news.google.com/rss/search?q=indian+stock+market+when:1d&hl=en-IN&gl=IN&ceid=IN:en",

]

def fetch\_rss\_articles(rss\_sources, max\_articles\_per\_feed=30):

all\_articles = []

for url in tqdm(rss\_sources, desc="Fetching RSS feeds"):

try:

feed = feedparser.parse(url)

for entry in feed.entries[:max\_articles\_per\_feed]:

all\_articles.append({

"stock": "",

"title": entry.get("title", "").strip(),

"description": entry.get("summary", entry.get("description", "")),

"pubDate": entry.get("published", entry.get("pubDate", "")),

"link": entry.get("link", ""),

"source": feed.feed.get("title", url)

})

except Exception as e:

print(f"❌ Error parsing {url}: {e}")

return all\_articles

all\_rss\_articles = fetch\_rss\_articles(rss\_sources, max\_articles\_per\_feed=30)

all\_rss\_df = pd.DataFrame(all\_rss\_articles)

# ✅ STEP 3: Merge ALL sources & Save CSV

all\_news = pd.concat([

newsdata\_df, google\_news\_df, yahoo\_news\_df, all\_rss\_df

], ignore\_index=True)

all\_news.drop\_duplicates(subset=['title', 'link'], inplace=True)

csv\_path = f"all\_news\_combined\_{datetime.now().strftime('%Y-%m-%d')}.csv"

all\_news.to\_csv(csv\_path, index=False)

print(f"\n✅ All news sources merged and saved as: {csv\_path}")

from IPython.display import display

display(all\_news.head(30))