017518709_PrithviElancherran_NLP_Project

December 8, 2024

[1]:
```
pip install yfinance pandas pandas-datareader
```

```
Requirement already satisfied: yfinance in /usr/local/lib/python3.10/dist-
packages (0.2.50)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages
(2.2.2)
Requirement already satisfied: pandas-datareader in
/usr/local/lib/python3.10/dist-packages (0.10.0)
Requirement already satisfied: numpy>=1.16.5 in /usr/local/lib/python3.10/dist-
packages (from yfinance) (1.26.4)
Requirement already satisfied: requests>=2.31 in /usr/local/lib/python3.10/dist-
packages (from yfinance) (2.32.3)
Requirement already satisfied: multitasking>=0.0.7 in
/usr/local/lib/python3.10/dist-packages (from yfinance) (0.0.11)
Requirement already satisfied: lxml>=4.9.1 in /usr/local/lib/python3.10/dist-
packages (from yfinance) (5.3.0)
Requirement already satisfied: platformdirs>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from yfinance) (4.3.6)
Requirement already satisfied: pytz>=2022.5 in /usr/local/lib/python3.10/dist-
packages (from yfinance) (2024.2)
Requirement already satisfied: frozendict>=2.3.4 in
/usr/local/lib/python3.10/dist-packages (from yfinance) (2.4.6)
Requirement already satisfied: peewee>=3.16.2 in /usr/local/lib/python3.10/dist-
packages (from yfinance) (3.17.8)
Requirement already satisfied: beautifulsoup4>=4.11.1 in
/usr/local/lib/python3.10/dist-packages (from yfinance) (4.12.3)
Requirement already satisfied: html5lib>=1.1 in /usr/local/lib/python3.10/dist-
packages (from yfinance) (1.1)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-
packages (from pandas) (2024.2)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-
packages (from beautifulsoup4>=4.11.1->yfinance) (2.6)
Requirement already satisfied: six>=1.9 in /usr/local/lib/python3.10/dist-
packages (from html5lib>=1.1->yfinance) (1.16.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-
packages (from html5lib>=1.1->yfinance) (0.5.1)
```

```
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests>=2.31->yfinance) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance)
(2024.8.30)
```

###Select list of 500 company stocks for which we will be preparing documents for RAG.

[2]:
```
import pandas as pd
import pandas_datareader as pdr

def fetch_sp500_tickers():
    # Fetch the S&P 500 table from Wikipedia
    table = pd.read_html('https://en.wikipedia.org/wiki/
 ↪List_of_S%26P_500_companies')
    sp500_df = table[0]
    tickers = sp500_df['Symbol'].tolist()
    # Adjust tickers for specific API requirements (e.g., replace '.' with '-')
    tickers = [ticker.replace('.', '-') for ticker in tickers]
    return tickers

sp500_tickers = fetch_sp500_tickers()
print(sp500_tickers)
```

```
['MMM', 'AOS', 'ABT', 'ABBV', 'ACN', 'ADBE', 'AMD', 'AES', 'AFL', 'A', 'APD',
'ABNB', 'AKAM', 'ALB', 'ARE', 'ALGN', 'ALLE', 'LNT', 'ALL', 'GOOGL', 'GOOG',
'MO', 'AMZN', 'AMCR', 'AMTM', 'AEE', 'AEP', 'AXP', 'AIG', 'AMT', 'AWK', 'AMP',
'AME', 'AMGN', 'APH', 'ADI', 'ANSS', 'AON', 'APA', 'AAPL', 'AMAT', 'APTV',
'ACGL', 'ADM', 'ANET', 'AJG', 'AIZ', 'T', 'ATO', 'ADSK', 'ADP', 'AZO', 'AVB',
'AVY', 'AXON', 'BKR', 'BALL', 'BAC', 'BAX', 'BDX', 'BRK-B', 'BBY', 'TECH',
'BIIB', 'BLK', 'BX', 'BK', 'BA', 'BKNG', 'BWA', 'BSX', 'BMY', 'AVGO', 'BR',
'BRO', 'BF-B', 'BLDR', 'BG', 'BXP', 'CHRW', 'CDNS', 'CZR', 'CPT', 'CPB', 'COF',
'CAH', 'KMX', 'CCL', 'CARR', 'CTLT', 'CAT', 'CBOE', 'CBRE', 'CDW', 'CE', 'COR',
'CNC', 'CNP', 'CF', 'CRL', 'SCHW', 'CHTR', 'CVX', 'CMG', 'CB', 'CHD', 'CI',
'CINF', 'CTAS', 'CSCO', 'C', 'CFG', 'CLX', 'CME', 'CMS', 'KO', 'CTSH', 'CL',
'CMCSA', 'CAG', 'COP', 'ED', 'STZ', 'CEG', 'COO', 'CPRT', 'GLW', 'CPAY', 'CTVA',
'CSGP', 'COST', 'CTRA', 'CRWD', 'CCI', 'CSX', 'CMI', 'CVS', 'DHR', 'DRI', 'DVA',
'DAY', 'DECK', 'DE', 'DELL', 'DAL', 'DVN', 'DXCM', 'FANG', 'DLR', 'DFS', 'DG',
'DLTR', 'D', 'DPZ', 'DOV', 'DOW', 'DHI', 'DTE', 'DUK', 'DD', 'EMN', 'ETN',
'EBAY', 'ECL', 'EIX', 'EW', 'EA', 'ELV', 'EMR', 'ENPH', 'ETR', 'EOG', 'EPAM',
'EQT', 'EFX', 'EQIX', 'EQR', 'ERIE', 'ESS', 'EL', 'EG', 'EVRG', 'ES', 'EXC',
'EXPE', 'EXPD', 'EXR', 'XOM', 'FFIV', 'FDS', 'FICO', 'FAST', 'FRT', 'FDX',
'FIS', 'FITB', 'FSLR', 'FE', 'FI', 'FMC', 'F', 'FTNT', 'FTV', 'FOXA', 'FOX',
'BEN', 'FCX', 'GRMN', 'IT', 'GE', 'GEHC', 'GEV', 'GEN', 'GNRC', 'GD', 'GIS',
```

```
'GM', 'GPC', 'GILD', 'GPN', 'GL', 'GDDY', 'GS', 'HAL', 'HIG', 'HAS', 'HCA',
'DOC', 'HSIC', 'HSY', 'HES', 'HPE', 'HLT', 'HOLX', 'HD', 'HON', 'HRL', 'HST',
'HWM', 'HPQ', 'HUBB', 'HUM', 'HBAN', 'HII', 'IBM', 'IEX', 'IDXX', 'ITW', 'INCY',
'IR', 'PODD', 'INTC', 'ICE', 'IFF', 'IP', 'IPG', 'INTU', 'ISRG', 'IVZ', 'INVH',
'IQV', 'IRM', 'JBHT', 'JBL', 'JKHY', 'J', 'JNJ', 'JCI', 'JPM', 'JNPR', 'K',
'KVUE', 'KDP', 'KEY', 'KEYS', 'KMB', 'KIM', 'KMI', 'KKR', 'KLAC', 'KHC', 'KR',
'LHX', 'LH', 'LRCX', 'LW', 'LVS', 'LDOS', 'LEN', 'LLY', 'LIN', 'LYV', 'LKQ',
'LMT', 'L', 'LOW', 'LULU', 'LYB', 'MTB', 'MPC', 'MKTX', 'MAR', 'MMC', 'MLM',
'MAS', 'MA', 'MTCH', 'MKC', 'MCD', 'MCK', 'MDT', 'MRK', 'META', 'MET', 'MTD',
'MGM', 'MCHP', 'MU', 'MSFT', 'MAA', 'MRNA', 'MHK', 'MOH', 'TAP', 'MDLZ', 'MPWR',
'MNST', 'MCO', 'MS', 'MOS', 'MSI', 'MSCI', 'NDAQ', 'NTAP', 'NFLX', 'NEM',
'NWSA', 'NWS', 'NEE', 'NKE', 'NI', 'NDSN', 'NSC', 'NTRS', 'NOC', 'NCLH', 'NRG',
'NUE', 'NVDA', 'NVR', 'NXPI', 'ORLY', 'OXY', 'ODFL', 'OMC', 'ON', 'OKE', 'ORCL',
'OTIS', 'PCAR', 'PKG', 'PLTR', 'PANW', 'PARA', 'PH', 'PAYX', 'PAYC', 'PYPL',
'PNR', 'PEP', 'PFE', 'PCG', 'PM', 'PSX', 'PNW', 'PNC', 'POOL', 'PPG', 'PPL',
'PFG', 'PG', 'PGR', 'PLD', 'PRU', 'PEG', 'PTC', 'PSA', 'PHM', 'QRVO', 'PWR',
'QCOM', 'DGX', 'RL', 'RJF', 'RTX', 'O', 'REG', 'REGN', 'RF', 'RSG', 'RMD',
'RVTY', 'ROK', 'ROL', 'ROP', 'ROST', 'RCL', 'SPGI', 'CRM', 'SBAC', 'SLB', 'STX',
'SRE', 'NOW', 'SHW', 'SPG', 'SWKS', 'SJM', 'SW', 'SNA', 'SOLV', 'SO', 'LUV',
'SWK', 'SBUX', 'STT', 'STLD', 'STE', 'SYK', 'SMCI', 'SYF', 'SNPS', 'SYY',
'TMUS', 'TROW', 'TTWO', 'TPR', 'TRGP', 'TGT', 'TEL', 'TDY', 'TFX', 'TER',
'TSLA', 'TXN', 'TPL', 'TXT', 'TMO', 'TJX', 'TSCO', 'TT', 'TDG', 'TRV', 'TRMB',
'TFC', 'TYL', 'TSN', 'USB', 'UBER', 'UDR', 'ULTA', 'UNP', 'UAL', 'UPS', 'URI',
'UNH', 'UHS', 'VLO', 'VTR', 'VLTO', 'VRSN', 'VRSK', 'VZ', 'VRTX', 'VTRS',
'VICI', 'V', 'VST', 'VMC', 'WRB', 'GWW', 'WAB', 'WBA', 'WMT', 'DIS', 'WBD',
'WM', 'WAT', 'WEC', 'WFC', 'WELL', 'WST', 'WDC', 'WY', 'WMB', 'WTW', 'WYNN',
'XEL', 'XYL', 'YUM', 'ZBRA', 'ZBH', 'ZTS']
```

#Preparing Data/Documents for RAG - From Yahoo Finance

[3]:
```python
import yfinance as yf
from datetime import datetime, timedelta

def download_ticker_data(tickers):
    end_date = datetime.now()
    start_date = end_date - timedelta(days=90)  # Using 3 months for data size␣
 ↪management
    data = {}
    # Attempt to download each ticker individually to handle failures
    for ticker in tickers:
        try:
            data[ticker] = yf.download(ticker, start=start_date, end=end_date)
            print(f"Downloaded data for {ticker}")
        except Exception as e:
            print(f"Failed to download data for {ticker}: {e}")
            # Retry logic can be added here if desired
    return data
```

```
# Fetch and download data
sp500_tickers = fetch_sp500_tickers()  # Assuming this function is defined␣
 ↪elsewhere
financial_data = download_ticker_data(sp500_tickers)
```

```
 [**********************100%**********************]  1 of 1 completed
Downloaded data for MMM
 [**********************100%**********************]  1 of 1 completed
Downloaded data for AOS
 [**********************100%**********************]  1 of 1 completed
Downloaded data for ABT
 [**********************100%**********************]  1 of 1 completed
Downloaded data for ABBV
 [**********************100%**********************]  1 of 1 completed
Downloaded data for ACN
 [**********************100%**********************]  1 of 1 completed
Downloaded data for ADBE
 [**********************100%**********************]  1 of 1 completed
Downloaded data for AMD
 [**********************100%**********************]  1 of 1 completed
Downloaded data for AES
 [**********************100%**********************]  1 of 1 completed
Downloaded data for AFL
 [**********************100%**********************]  1 of 1 completed
Downloaded data for A
 [**********************100%**********************]  1 of 1 completed
Downloaded data for APD
 [**********************100%**********************]  1 of 1 completed
Downloaded data for ABNB
 [**********************100%**********************]  1 of 1 completed
Downloaded data for AKAM
 [**********************100%**********************]  1 of 1 completed
Downloaded data for ALB
```

```
 [**********100%**********]  1 of 1 completed
Downloaded data for ARE
 [**********100%**********]  1 of 1 completed
Downloaded data for ALGN
 [**********100%**********]  1 of 1 completed
Downloaded data for ALLE
 [**********100%**********]  1 of 1 completed
Downloaded data for LNT
 [**********100%**********]  1 of 1 completed
Downloaded data for ALL
 [**********100%**********]  1 of 1 completed
Downloaded data for GOOGL
 [**********100%**********]  1 of 1 completed
Downloaded data for GOOG
 [**********100%**********]  1 of 1 completed
Downloaded data for MO
 [**********100%**********]  1 of 1 completed
Downloaded data for AMZN
 [**********100%**********]  1 of 1 completed
Downloaded data for AMCR
 [**********100%**********]  1 of 1 completed
Downloaded data for AMTM
 [**********100%**********]  1 of 1 completed
Downloaded data for AEE
 [**********100%**********]  1 of 1 completed
Downloaded data for AEP
 [**********100%**********]  1 of 1 completed
Downloaded data for AXP
 [**********100%**********]  1 of 1 completed
Downloaded data for AIG
 [**********100%**********]  1 of 1 completed
Downloaded data for AMT
```

```
 [**********100%**********]  1 of 1 completed
Downloaded data for AWK
 [**********100%**********]  1 of 1 completed
Downloaded data for AMP
 [**********100%**********]  1 of 1 completed
Downloaded data for AME
 [**********100%**********]  1 of 1 completed
Downloaded data for AMGN
 [**********100%**********]  1 of 1 completed
Downloaded data for APH
 [**********100%**********]  1 of 1 completed
Downloaded data for ADI
 [**********100%**********]  1 of 1 completed
Downloaded data for ANSS
 [**********100%**********]  1 of 1 completed
Downloaded data for AON
 [**********100%**********]  1 of 1 completed
Downloaded data for APA
 [**********100%**********]  1 of 1 completed
Downloaded data for AAPL
 [**********100%**********]  1 of 1 completed
Downloaded data for AMAT
 [**********100%**********]  1 of 1 completed
Downloaded data for APTV
 [**********100%**********]  1 of 1 completed
Downloaded data for ACGL
 [**********100%**********]  1 of 1 completed
Downloaded data for ADM
 [**********100%**********]  1 of 1 completed
Downloaded data for ANET
 [**********100%**********]  1 of 1 completed
Downloaded data for AJG
```

```
 [**********************100%**********************]  1 of 1 completed
Downloaded data for AIZ
 [**********************100%**********************]  1 of 1 completed
Downloaded data for T
 [**********************100%**********************]  1 of 1 completed
Downloaded data for ATO
 [**********************100%**********************]  1 of 1 completed
Downloaded data for ADSK
 [**********************100%**********************]  1 of 1 completed
Downloaded data for ADP
 [**********************100%**********************]  1 of 1 completed
Downloaded data for AZO
 [**********************100%**********************]  1 of 1 completed
Downloaded data for AVB
 [**********************100%**********************]  1 of 1 completed
Downloaded data for AVY
 [**********************100%**********************]  1 of 1 completed
Downloaded data for AXON
 [**********************100%**********************]  1 of 1 completed
Downloaded data for BKR
 [**********************100%**********************]  1 of 1 completed
Downloaded data for BALL
 [**********************100%**********************]  1 of 1 completed
Downloaded data for BAC
 [**********************100%**********************]  1 of 1 completed
Downloaded data for BAX
 [**********************100%**********************]  1 of 1 completed
Downloaded data for BDX
 [**********************100%**********************]  1 of 1 completed
Downloaded data for BRK-B
 [**********************100%**********************]  1 of 1 completed
Downloaded data for BBY
```

```
 [**********************100%**********************]  1 of 1 completed
Downloaded data for TECH
 [**********************100%**********************]  1 of 1 completed
Downloaded data for BIIB
 [**********************100%**********************]  1 of 1 completed
Downloaded data for BLK
 [**********************100%**********************]  1 of 1 completed
Downloaded data for BX
 [**********************100%**********************]  1 of 1 completed
Downloaded data for BK
 [**********************100%**********************]  1 of 1 completed
Downloaded data for BA
 [**********************100%**********************]  1 of 1 completed
Downloaded data for BKNG
 [**********************100%**********************]  1 of 1 completed
Downloaded data for BWA
 [**********************100%**********************]  1 of 1 completed
Downloaded data for BSX
 [**********************100%**********************]  1 of 1 completed
Downloaded data for BMY
 [**********************100%**********************]  1 of 1 completed
Downloaded data for AVGO
 [**********************100%**********************]  1 of 1 completed
Downloaded data for BR
 [**********************100%**********************]  1 of 1 completed
Downloaded data for BRO
 [**********************100%**********************]  1 of 1 completed
Downloaded data for BF-B
 [**********************100%**********************]  1 of 1 completed
Downloaded data for BLDR
 [**********************100%**********************]  1 of 1 completed
Downloaded data for BG
```

```
 [**********100%**********]  1 of 1 completed
Downloaded data for BXP
 [**********100%**********]  1 of 1 completed
Downloaded data for CHRW
 [**********100%**********]  1 of 1 completed
Downloaded data for CDNS
 [**********100%**********]  1 of 1 completed
Downloaded data for CZR
 [**********100%**********]  1 of 1 completed
Downloaded data for CPT
 [**********100%**********]  1 of 1 completed
Downloaded data for CPB
 [**********100%**********]  1 of 1 completed
Downloaded data for COF
 [**********100%**********]  1 of 1 completed
Downloaded data for CAH
 [**********100%**********]  1 of 1 completed
Downloaded data for KMX
 [**********100%**********]  1 of 1 completed
Downloaded data for CCL
 [**********100%**********]  1 of 1 completed
Downloaded data for CARR
 [**********100%**********]  1 of 1 completed
Downloaded data for CTLT
 [**********100%**********]  1 of 1 completed
Downloaded data for CAT
 [**********100%**********]  1 of 1 completed
Downloaded data for CBOE
 [**********100%**********]  1 of 1 completed
Downloaded data for CBRE
 [**********100%**********]  1 of 1 completed
Downloaded data for CDW
```

```
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CE
 [**********************100%**********************]  1 of 1 completed
Downloaded data for COR
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CNC
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CNP
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CF
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CRL
 [**********************100%**********************]  1 of 1 completed
Downloaded data for SCHW
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CHTR
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CVX
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CMG
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CB
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CHD
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CI
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CINF
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CTAS
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CSCO
```

```
 [**********************100%**********************]  1 of 1 completed
Downloaded data for C
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CFG
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CLX
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CME
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CMS
 [**********************100%**********************]  1 of 1 completed
Downloaded data for KO
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CTSH
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CL
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CMCSA
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CAG
 [**********************100%**********************]  1 of 1 completed
Downloaded data for COP
 [**********************100%**********************]  1 of 1 completed
Downloaded data for ED
 [**********************100%**********************]  1 of 1 completed
Downloaded data for STZ
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CEG
 [**********************100%**********************]  1 of 1 completed
Downloaded data for COO
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CPRT
```

```
 [**********************100%**********************]  1 of 1 completed
Downloaded data for GLW
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CPAY
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CTVA
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CSGP
 [**********************100%**********************]  1 of 1 completed
Downloaded data for COST
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CTRA
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CRWD
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CCI
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CSX
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CMI
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CVS
 [**********************100%**********************]  1 of 1 completed
Downloaded data for DHR
 [**********************100%**********************]  1 of 1 completed
Downloaded data for DRI
 [**********************100%**********************]  1 of 1 completed
Downloaded data for DVA
 [**********************100%**********************]  1 of 1 completed
Downloaded data for DAY
 [**********************100%**********************]  1 of 1 completed
Downloaded data for DECK
```

```
 [**********************100%**********************]  1 of 1 completed
Downloaded data for DE
 [**********************100%**********************]  1 of 1 completed
Downloaded data for DELL
 [**********************100%**********************]  1 of 1 completed
Downloaded data for DAL
 [**********************100%**********************]  1 of 1 completed
Downloaded data for DVN
 [**********************100%**********************]  1 of 1 completed
Downloaded data for DXCM
 [**********************100%**********************]  1 of 1 completed
Downloaded data for FANG
 [**********************100%**********************]  1 of 1 completed
Downloaded data for DLR
 [**********************100%**********************]  1 of 1 completed
Downloaded data for DFS
 [**********************100%**********************]  1 of 1 completed
Downloaded data for DG
 [**********************100%**********************]  1 of 1 completed
Downloaded data for DLTR
 [**********************100%**********************]  1 of 1 completed
Downloaded data for D
 [**********************100%**********************]  1 of 1 completed
Downloaded data for DPZ
 [**********************100%**********************]  1 of 1 completed
Downloaded data for DOV
 [**********************100%**********************]  1 of 1 completed
Downloaded data for DOW
 [**********************100%**********************]  1 of 1 completed
Downloaded data for DHI
 [**********************100%**********************]  1 of 1 completed
Downloaded data for DTE
```

```
 [**********100%**********]  1 of 1 completed
Downloaded data for DUK
 [**********100%**********]  1 of 1 completed
Downloaded data for DD
 [**********100%**********]  1 of 1 completed
Downloaded data for EMN
 [**********100%**********]  1 of 1 completed
Downloaded data for ETN
 [**********100%**********]  1 of 1 completed
Downloaded data for EBAY
 [**********100%**********]  1 of 1 completed
Downloaded data for ECL
 [**********100%**********]  1 of 1 completed
Downloaded data for EIX
 [**********100%**********]  1 of 1 completed
Downloaded data for EW
 [**********100%**********]  1 of 1 completed
Downloaded data for EA
 [**********100%**********]  1 of 1 completed
Downloaded data for ELV
 [**********100%**********]  1 of 1 completed
Downloaded data for EMR
 [**********100%**********]  1 of 1 completed
Downloaded data for ENPH
 [**********100%**********]  1 of 1 completed
Downloaded data for ETR
 [**********100%**********]  1 of 1 completed
Downloaded data for EOG
 [**********100%**********]  1 of 1 completed
Downloaded data for EPAM
 [**********100%**********]  1 of 1 completed
Downloaded data for EQT
```

```
 [**********100%**********]  1 of 1 completed
Downloaded data for EFX
 [**********100%**********]  1 of 1 completed
Downloaded data for EQIX
 [**********100%**********]  1 of 1 completed
Downloaded data for EQR
 [**********100%**********]  1 of 1 completed
Downloaded data for ERIE
 [**********100%**********]  1 of 1 completed
Downloaded data for ESS
 [**********100%**********]  1 of 1 completed
Downloaded data for EL
 [**********100%**********]  1 of 1 completed
Downloaded data for EG
 [**********100%**********]  1 of 1 completed
Downloaded data for EVRG
 [**********100%**********]  1 of 1 completed
Downloaded data for ES
 [**********100%**********]  1 of 1 completed
Downloaded data for EXC
 [**********100%**********]  1 of 1 completed
Downloaded data for EXPE
 [**********100%**********]  1 of 1 completed
Downloaded data for EXPD
 [**********100%**********]  1 of 1 completed
Downloaded data for EXR
 [**********100%**********]  1 of 1 completed
Downloaded data for XOM
 [**********100%**********]  1 of 1 completed
Downloaded data for FFIV
 [**********100%**********]  1 of 1 completed
Downloaded data for FDS
```

```
 [**********************100%**********************]  1 of 1 completed
Downloaded data for FICO
 [**********************100%**********************]  1 of 1 completed
Downloaded data for FAST
 [**********************100%**********************]  1 of 1 completed
Downloaded data for FRT
 [**********************100%**********************]  1 of 1 completed
Downloaded data for FDX
 [**********************100%**********************]  1 of 1 completed
Downloaded data for FIS
 [**********************100%**********************]  1 of 1 completed
Downloaded data for FITB
 [**********************100%**********************]  1 of 1 completed
Downloaded data for FSLR
 [**********************100%**********************]  1 of 1 completed
Downloaded data for FE
 [**********************100%**********************]  1 of 1 completed
Downloaded data for FI
 [**********************100%**********************]  1 of 1 completed
Downloaded data for FMC
 [**********************100%**********************]  1 of 1 completed
Downloaded data for F
 [**********************100%**********************]  1 of 1 completed
Downloaded data for FTNT
 [**********************100%**********************]  1 of 1 completed
Downloaded data for FTV
 [**********************100%**********************]  1 of 1 completed
Downloaded data for FOXA
 [**********************100%**********************]  1 of 1 completed
Downloaded data for FOX
 [**********************100%**********************]  1 of 1 completed
Downloaded data for BEN
```

```
 [**********100%**********]  1 of 1 completed
Downloaded data for FCX
 [**********100%**********]  1 of 1 completed
Downloaded data for GRMN
 [**********100%**********]  1 of 1 completed
Downloaded data for IT
 [**********100%**********]  1 of 1 completed
Downloaded data for GE
 [**********100%**********]  1 of 1 completed
Downloaded data for GEHC
 [**********100%**********]  1 of 1 completed
Downloaded data for GEV
 [**********100%**********]  1 of 1 completed
Downloaded data for GEN
 [**********100%**********]  1 of 1 completed
Downloaded data for GNRC
 [**********100%**********]  1 of 1 completed
Downloaded data for GD
 [**********100%**********]  1 of 1 completed
Downloaded data for GIS
 [**********100%**********]  1 of 1 completed
Downloaded data for GM
 [**********100%**********]  1 of 1 completed
Downloaded data for GPC
 [**********100%**********]  1 of 1 completed
Downloaded data for GILD
 [**********100%**********]  1 of 1 completed
Downloaded data for GPN
 [**********100%**********]  1 of 1 completed
Downloaded data for GL
 [**********100%**********]  1 of 1 completed
Downloaded data for GDDY
```

```
 [**********************100%**********************]  1 of 1 completed
Downloaded data for GS
 [**********************100%**********************]  1 of 1 completed
Downloaded data for HAL
 [**********************100%**********************]  1 of 1 completed
Downloaded data for HIG
 [**********************100%**********************]  1 of 1 completed
Downloaded data for HAS
 [**********************100%**********************]  1 of 1 completed
Downloaded data for HCA
 [**********************100%**********************]  1 of 1 completed
Downloaded data for DOC
 [**********************100%**********************]  1 of 1 completed
Downloaded data for HSIC
 [**********************100%**********************]  1 of 1 completed
Downloaded data for HSY
 [**********************100%**********************]  1 of 1 completed
Downloaded data for HES
 [**********************100%**********************]  1 of 1 completed
Downloaded data for HPE
 [**********************100%**********************]  1 of 1 completed
Downloaded data for HLT
 [**********************100%**********************]  1 of 1 completed
Downloaded data for HOLX
 [**********************100%**********************]  1 of 1 completed
Downloaded data for HD
 [**********************100%**********************]  1 of 1 completed
Downloaded data for HON
 [**********************100%**********************]  1 of 1 completed
Downloaded data for HRL
 [**********************100%**********************]  1 of 1 completed
Downloaded data for HST
```

```
 [**********************100%**********************]  1 of 1 completed
Downloaded data for HWM
 [**********************100%**********************]  1 of 1 completed
Downloaded data for HPQ
 [**********************100%**********************]  1 of 1 completed
Downloaded data for HUBB
 [**********************100%**********************]  1 of 1 completed
Downloaded data for HUM
 [**********************100%**********************]  1 of 1 completed
Downloaded data for HBAN
 [**********************100%**********************]  1 of 1 completed
Downloaded data for HII
 [**********************100%**********************]  1 of 1 completed
Downloaded data for IBM
 [**********************100%**********************]  1 of 1 completed
Downloaded data for IEX
 [**********************100%**********************]  1 of 1 completed
Downloaded data for IDXX
 [**********************100%**********************]  1 of 1 completed
Downloaded data for ITW
 [**********************100%**********************]  1 of 1 completed
Downloaded data for INCY
 [**********************100%**********************]  1 of 1 completed
Downloaded data for IR
 [**********************100%**********************]  1 of 1 completed
Downloaded data for PODD
 [**********************100%**********************]  1 of 1 completed
Downloaded data for INTC
 [**********************100%**********************]  1 of 1 completed
Downloaded data for ICE
 [**********************100%**********************]  1 of 1 completed
Downloaded data for IFF
```

```
 [**********************100%**********************]  1 of 1 completed
Downloaded data for IP
 [**********************100%**********************]  1 of 1 completed
Downloaded data for IPG
 [**********************100%**********************]  1 of 1 completed
Downloaded data for INTU
 [**********************100%**********************]  1 of 1 completed
Downloaded data for ISRG
 [**********************100%**********************]  1 of 1 completed
Downloaded data for IVZ
 [**********************100%**********************]  1 of 1 completed
Downloaded data for INVH
 [**********************100%**********************]  1 of 1 completed
Downloaded data for IQV
 [**********************100%**********************]  1 of 1 completed
Downloaded data for IRM
 [**********************100%**********************]  1 of 1 completed
Downloaded data for JBHT
 [**********************100%**********************]  1 of 1 completed
Downloaded data for JBL
 [**********************100%**********************]  1 of 1 completed
Downloaded data for JKHY
 [**********************100%**********************]  1 of 1 completed
Downloaded data for J
 [**********************100%**********************]  1 of 1 completed
Downloaded data for JNJ
 [**********************100%**********************]  1 of 1 completed
Downloaded data for JCI
 [**********************100%**********************]  1 of 1 completed
Downloaded data for JPM
 [**********************100%**********************]  1 of 1 completed
Downloaded data for JNPR
```

```
 [**********************100%**********************]  1 of 1 completed
Downloaded data for K
 [**********************100%**********************]  1 of 1 completed
Downloaded data for KVUE
 [**********************100%**********************]  1 of 1 completed
Downloaded data for KDP
 [**********************100%**********************]  1 of 1 completed
Downloaded data for KEY
 [**********************100%**********************]  1 of 1 completed
Downloaded data for KEYS
 [**********************100%**********************]  1 of 1 completed
Downloaded data for KMB
 [**********************100%**********************]  1 of 1 completed
Downloaded data for KIM
 [**********************100%**********************]  1 of 1 completed
Downloaded data for KMI
 [**********************100%**********************]  1 of 1 completed
Downloaded data for KKR
 [**********************100%**********************]  1 of 1 completed
Downloaded data for KLAC
 [**********************100%**********************]  1 of 1 completed
Downloaded data for KHC
 [**********************100%**********************]  1 of 1 completed
Downloaded data for KR
 [**********************100%**********************]  1 of 1 completed
Downloaded data for LHX
 [**********************100%**********************]  1 of 1 completed
Downloaded data for LH
 [**********************100%**********************]  1 of 1 completed
Downloaded data for LRCX
 [**********************100%**********************]  1 of 1 completed
Downloaded data for LW
```

```
 [**********100%**********]  1 of 1 completed
Downloaded data for LVS
 [**********100%**********]  1 of 1 completed
Downloaded data for LDOS
 [**********100%**********]  1 of 1 completed
Downloaded data for LEN
 [**********100%**********]  1 of 1 completed
Downloaded data for LLY
 [**********100%**********]  1 of 1 completed
Downloaded data for LIN
 [**********100%**********]  1 of 1 completed
Downloaded data for LYV
 [**********100%**********]  1 of 1 completed
Downloaded data for LKQ
 [**********100%**********]  1 of 1 completed
Downloaded data for LMT
 [**********100%**********]  1 of 1 completed
Downloaded data for L
 [**********100%**********]  1 of 1 completed
Downloaded data for LOW
 [**********100%**********]  1 of 1 completed
Downloaded data for LULU
 [**********100%**********]  1 of 1 completed
Downloaded data for LYB
 [**********100%**********]  1 of 1 completed
Downloaded data for MTB
 [**********100%**********]  1 of 1 completed
Downloaded data for MPC
 [**********100%**********]  1 of 1 completed
Downloaded data for MKTX
 [**********100%**********]  1 of 1 completed
Downloaded data for MAR
```

```
 [**********100%**********]  1 of 1 completed
Downloaded data for MMC
 [**********100%**********]  1 of 1 completed
Downloaded data for MLM
 [**********100%**********]  1 of 1 completed
Downloaded data for MAS
 [**********100%**********]  1 of 1 completed
Downloaded data for MA
 [**********100%**********]  1 of 1 completed
Downloaded data for MTCH
 [**********100%**********]  1 of 1 completed
Downloaded data for MKC
 [**********100%**********]  1 of 1 completed
Downloaded data for MCD
 [**********100%**********]  1 of 1 completed
Downloaded data for MCK
 [**********100%**********]  1 of 1 completed
Downloaded data for MDT
 [**********100%**********]  1 of 1 completed
Downloaded data for MRK
 [**********100%**********]  1 of 1 completed
Downloaded data for META
 [**********100%**********]  1 of 1 completed
Downloaded data for MET
 [**********100%**********]  1 of 1 completed
Downloaded data for MTD
 [**********100%**********]  1 of 1 completed
Downloaded data for MGM
 [**********100%**********]  1 of 1 completed
Downloaded data for MCHP
 [**********100%**********]  1 of 1 completed
Downloaded data for MU
```

```
 [**********************100%**********************]  1 of 1 completed
Downloaded data for MSFT
 [**********************100%**********************]  1 of 1 completed
Downloaded data for MAA
 [**********************100%**********************]  1 of 1 completed
Downloaded data for MRNA
 [**********************100%**********************]  1 of 1 completed
Downloaded data for MHK
 [**********************100%**********************]  1 of 1 completed
Downloaded data for MOH
 [**********************100%**********************]  1 of 1 completed
Downloaded data for TAP
 [**********************100%**********************]  1 of 1 completed
Downloaded data for MDLZ
 [**********************100%**********************]  1 of 1 completed
Downloaded data for MPWR
 [**********************100%**********************]  1 of 1 completed
Downloaded data for MNST
 [**********************100%**********************]  1 of 1 completed
Downloaded data for MCO
 [**********************100%**********************]  1 of 1 completed
Downloaded data for MS
 [**********************100%**********************]  1 of 1 completed
Downloaded data for MOS
 [**********************100%**********************]  1 of 1 completed
Downloaded data for MSI
 [**********************100%**********************]  1 of 1 completed
Downloaded data for MSCI
 [**********************100%**********************]  1 of 1 completed
Downloaded data for NDAQ
 [**********************100%**********************]  1 of 1 completed
Downloaded data for NTAP
```

```
 [**********************100%**********************]  1 of 1 completed
Downloaded data for NFLX
 [**********************100%**********************]  1 of 1 completed
Downloaded data for NEM
 [**********************100%**********************]  1 of 1 completed
Downloaded data for NWSA
 [**********************100%**********************]  1 of 1 completed
Downloaded data for NWS
 [**********************100%**********************]  1 of 1 completed
Downloaded data for NEE
 [**********************100%**********************]  1 of 1 completed
Downloaded data for NKE
 [**********************100%**********************]  1 of 1 completed
Downloaded data for NI
 [**********************100%**********************]  1 of 1 completed
Downloaded data for NDSN
 [**********************100%**********************]  1 of 1 completed
Downloaded data for NSC
 [**********************100%**********************]  1 of 1 completed
Downloaded data for NTRS
 [**********************100%**********************]  1 of 1 completed
Downloaded data for NOC
 [**********************100%**********************]  1 of 1 completed
Downloaded data for NCLH
 [**********************100%**********************]  1 of 1 completed
Downloaded data for NRG
 [**********************100%**********************]  1 of 1 completed
Downloaded data for NUE
 [**********************100%**********************]  1 of 1 completed
Downloaded data for NVDA
 [**********************100%**********************]  1 of 1 completed
Downloaded data for NVR
```

```
 [**********100%**********]  1 of 1 completed
Downloaded data for NXPI
 [**********100%**********]  1 of 1 completed
Downloaded data for ORLY
 [**********100%**********]  1 of 1 completed
Downloaded data for OXY
 [**********100%**********]  1 of 1 completed
Downloaded data for ODFL
 [**********100%**********]  1 of 1 completed
Downloaded data for OMC
 [**********100%**********]  1 of 1 completed
Downloaded data for ON
 [**********100%**********]  1 of 1 completed
Downloaded data for OKE
 [**********100%**********]  1 of 1 completed
Downloaded data for ORCL
 [**********100%**********]  1 of 1 completed
Downloaded data for OTIS
 [**********100%**********]  1 of 1 completed
Downloaded data for PCAR
 [**********100%**********]  1 of 1 completed
Downloaded data for PKG
 [**********100%**********]  1 of 1 completed
Downloaded data for PLTR
 [**********100%**********]  1 of 1 completed
Downloaded data for PANW
 [**********100%**********]  1 of 1 completed
Downloaded data for PARA
 [**********100%**********]  1 of 1 completed
Downloaded data for PH
 [**********100%**********]  1 of 1 completed
Downloaded data for PAYX
```

```
 [*********************100%***********************]  1 of 1 completed
Downloaded data for PAYC
 [*********************100%***********************]  1 of 1 completed
Downloaded data for PYPL
 [*********************100%***********************]  1 of 1 completed
Downloaded data for PNR
 [*********************100%***********************]  1 of 1 completed
Downloaded data for PEP
 [*********************100%***********************]  1 of 1 completed
Downloaded data for PFE
 [*********************100%***********************]  1 of 1 completed
Downloaded data for PCG
 [*********************100%***********************]  1 of 1 completed
Downloaded data for PM
 [*********************100%***********************]  1 of 1 completed
Downloaded data for PSX
 [*********************100%***********************]  1 of 1 completed
Downloaded data for PNW
 [*********************100%***********************]  1 of 1 completed
Downloaded data for PNC
 [*********************100%***********************]  1 of 1 completed
Downloaded data for POOL
 [*********************100%***********************]  1 of 1 completed
Downloaded data for PPG
 [*********************100%***********************]  1 of 1 completed
Downloaded data for PPL
 [*********************100%***********************]  1 of 1 completed
Downloaded data for PFG
 [*********************100%***********************]  1 of 1 completed
Downloaded data for PG
 [*********************100%***********************]  1 of 1 completed
Downloaded data for PGR
```

```
 [**********************100%**********************]  1 of 1 completed
Downloaded data for PLD
 [**********************100%**********************]  1 of 1 completed
Downloaded data for PRU
 [**********************100%**********************]  1 of 1 completed
Downloaded data for PEG
 [**********************100%**********************]  1 of 1 completed
Downloaded data for PTC
 [**********************100%**********************]  1 of 1 completed
Downloaded data for PSA
 [**********************100%**********************]  1 of 1 completed
Downloaded data for PHM
 [**********************100%**********************]  1 of 1 completed
Downloaded data for QRVO
 [**********************100%**********************]  1 of 1 completed
Downloaded data for PWR
 [**********************100%**********************]  1 of 1 completed
Downloaded data for QCOM
 [**********************100%**********************]  1 of 1 completed
Downloaded data for DGX
 [**********************100%**********************]  1 of 1 completed
Downloaded data for RL
 [**********************100%**********************]  1 of 1 completed
Downloaded data for RJF
 [**********************100%**********************]  1 of 1 completed
Downloaded data for RTX
 [**********************100%**********************]  1 of 1 completed
Downloaded data for O
 [**********************100%**********************]  1 of 1 completed
Downloaded data for REG
 [**********************100%**********************]  1 of 1 completed
Downloaded data for REGN
```

```
 [**********************100%**********************]  1 of 1 completed
Downloaded data for RF
 [**********************100%**********************]  1 of 1 completed
Downloaded data for RSG
 [**********************100%**********************]  1 of 1 completed
Downloaded data for RMD
 [**********************100%**********************]  1 of 1 completed
Downloaded data for RVTY
 [**********************100%**********************]  1 of 1 completed
Downloaded data for ROK
 [**********************100%**********************]  1 of 1 completed
Downloaded data for ROL
 [**********************100%**********************]  1 of 1 completed
Downloaded data for ROP
 [**********************100%**********************]  1 of 1 completed
Downloaded data for ROST
 [**********************100%**********************]  1 of 1 completed
Downloaded data for RCL
 [**********************100%**********************]  1 of 1 completed
Downloaded data for SPGI
 [**********************100%**********************]  1 of 1 completed
Downloaded data for CRM
 [**********************100%**********************]  1 of 1 completed
Downloaded data for SBAC
 [**********************100%**********************]  1 of 1 completed
Downloaded data for SLB
 [**********************100%**********************]  1 of 1 completed
Downloaded data for STX
 [**********************100%**********************]  1 of 1 completed
Downloaded data for SRE
 [**********************100%**********************]  1 of 1 completed
Downloaded data for NOW
```

```
 [**********************100%**********************]  1 of 1 completed
Downloaded data for SHW
 [**********************100%**********************]  1 of 1 completed
Downloaded data for SPG
 [**********************100%**********************]  1 of 1 completed
Downloaded data for SWKS
 [**********************100%**********************]  1 of 1 completed
Downloaded data for SJM
 [**********************100%**********************]  1 of 1 completed
Downloaded data for SW
 [**********************100%**********************]  1 of 1 completed
Downloaded data for SNA
 [**********************100%**********************]  1 of 1 completed
Downloaded data for SOLV
 [**********************100%**********************]  1 of 1 completed
Downloaded data for SO
 [**********************100%**********************]  1 of 1 completed
Downloaded data for LUV
 [**********************100%**********************]  1 of 1 completed
Downloaded data for SWK
 [**********************100%**********************]  1 of 1 completed
Downloaded data for SBUX
 [**********************100%**********************]  1 of 1 completed
Downloaded data for STT
 [**********************100%**********************]  1 of 1 completed
Downloaded data for STLD
 [**********************100%**********************]  1 of 1 completed
Downloaded data for STE
 [**********************100%**********************]  1 of 1 completed
Downloaded data for SYK
 [**********************100%**********************]  1 of 1 completed
Downloaded data for SMCI
```

```
 [**********************100%**********************]  1 of 1 completed
Downloaded data for SYF
 [**********************100%**********************]  1 of 1 completed
Downloaded data for SNPS
 [**********************100%**********************]  1 of 1 completed
Downloaded data for SYY
 [**********************100%**********************]  1 of 1 completed
Downloaded data for TMUS
 [**********************100%**********************]  1 of 1 completed
Downloaded data for TROW
 [**********************100%**********************]  1 of 1 completed
Downloaded data for TTWO
 [**********************100%**********************]  1 of 1 completed
Downloaded data for TPR
 [**********************100%**********************]  1 of 1 completed
Downloaded data for TRGP
 [**********************100%**********************]  1 of 1 completed
Downloaded data for TGT
 [**********************100%**********************]  1 of 1 completed
Downloaded data for TEL
 [**********************100%**********************]  1 of 1 completed
Downloaded data for TDY
 [**********************100%**********************]  1 of 1 completed
Downloaded data for TFX
 [**********************100%**********************]  1 of 1 completed
Downloaded data for TER
 [**********************100%**********************]  1 of 1 completed
Downloaded data for TSLA
 [**********************100%**********************]  1 of 1 completed
Downloaded data for TXN
 [**********************100%**********************]  1 of 1 completed
Downloaded data for TPL
```

```
 [**********100%**********]  1 of 1 completed
Downloaded data for TXT
 [**********100%**********]  1 of 1 completed
Downloaded data for TMO
 [**********100%**********]  1 of 1 completed
Downloaded data for TJX
 [**********100%**********]  1 of 1 completed
Downloaded data for TSCO
 [**********100%**********]  1 of 1 completed
Downloaded data for TT
 [**********100%**********]  1 of 1 completed
Downloaded data for TDG
 [**********100%**********]  1 of 1 completed
Downloaded data for TRV
 [**********100%**********]  1 of 1 completed
Downloaded data for TRMB
 [**********100%**********]  1 of 1 completed
Downloaded data for TFC
 [**********100%**********]  1 of 1 completed
Downloaded data for TYL
 [**********100%**********]  1 of 1 completed
Downloaded data for TSN
 [**********100%**********]  1 of 1 completed
Downloaded data for USB
 [**********100%**********]  1 of 1 completed
Downloaded data for UBER
 [**********100%**********]  1 of 1 completed
Downloaded data for UDR
 [**********100%**********]  1 of 1 completed
Downloaded data for ULTA
 [**********100%**********]  1 of 1 completed
Downloaded data for UNP
```

```
 [**********100%**********]  1 of 1 completed
Downloaded data for UAL
 [**********100%**********]  1 of 1 completed
Downloaded data for UPS
 [**********100%**********]  1 of 1 completed
Downloaded data for URI
 [**********100%**********]  1 of 1 completed
Downloaded data for UNH
 [**********100%**********]  1 of 1 completed
Downloaded data for UHS
 [**********100%**********]  1 of 1 completed
Downloaded data for VLO
 [**********100%**********]  1 of 1 completed
Downloaded data for VTR
 [**********100%**********]  1 of 1 completed
Downloaded data for VLTO
 [**********100%**********]  1 of 1 completed
Downloaded data for VRSN
 [**********100%**********]  1 of 1 completed
Downloaded data for VRSK
 [**********100%**********]  1 of 1 completed
Downloaded data for VZ
 [**********100%**********]  1 of 1 completed
Downloaded data for VRTX
 [**********100%**********]  1 of 1 completed
Downloaded data for VTRS
 [**********100%**********]  1 of 1 completed
Downloaded data for VICI
 [**********100%**********]  1 of 1 completed
Downloaded data for V
 [**********100%**********]  1 of 1 completed
Downloaded data for VST
```

```
 [**********************100%**********************]  1 of 1 completed
Downloaded data for VMC
 [**********************100%**********************]  1 of 1 completed
Downloaded data for WRB
 [**********************100%**********************]  1 of 1 completed
Downloaded data for GWW
 [**********************100%**********************]  1 of 1 completed
Downloaded data for WAB
 [**********************100%**********************]  1 of 1 completed
Downloaded data for WBA
 [**********************100%**********************]  1 of 1 completed
Downloaded data for WMT
 [**********************100%**********************]  1 of 1 completed
Downloaded data for DIS
 [**********************100%**********************]  1 of 1 completed
Downloaded data for WBD
 [**********************100%**********************]  1 of 1 completed
Downloaded data for WM
 [**********************100%**********************]  1 of 1 completed
Downloaded data for WAT
 [**********************100%**********************]  1 of 1 completed
Downloaded data for WEC
 [**********************100%**********************]  1 of 1 completed
Downloaded data for WFC
 [**********************100%**********************]  1 of 1 completed
Downloaded data for WELL
 [**********************100%**********************]  1 of 1 completed
Downloaded data for WST
 [**********************100%**********************]  1 of 1 completed
Downloaded data for WDC
 [**********************100%**********************]  1 of 1 completed
Downloaded data for WY
```

```
[**********100%**********]  1 of 1 completed
Downloaded data for WMB
[**********100%**********]  1 of 1 completed
Downloaded data for WTW
[**********100%**********]  1 of 1 completed
Downloaded data for WYNN
[**********100%**********]  1 of 1 completed
Downloaded data for XEL
[**********100%**********]  1 of 1 completed
Downloaded data for XYL
[**********100%**********]  1 of 1 completed
Downloaded data for YUM
[**********100%**********]  1 of 1 completed
Downloaded data for ZBRA
[**********100%**********]  1 of 1 completed
Downloaded data for ZBH
[**********100%**********]  1 of 1 completed
Downloaded data for ZTS
```

[4]:
```python
import os

def save_financial_data(data, folder="financial_data"):
    if not os.path.exists(folder):
        os.makedirs(folder)
    for ticker, df in data.items():
        file_path = os.path.join(folder, f"{ticker}.csv")
        df.to_csv(file_path)
        print(f"Saved data for {ticker} at {file_path}")

save_financial_data(financial_data)
```

```
Saved data for MMM at financial_data/MMM.csv
Saved data for AOS at financial_data/AOS.csv
Saved data for ABT at financial_data/ABT.csv
Saved data for ABBV at financial_data/ABBV.csv
Saved data for ACN at financial_data/ACN.csv
Saved data for ADBE at financial_data/ADBE.csv
Saved data for AMD at financial_data/AMD.csv
```

```
Saved data for AES at financial_data/AES.csv
Saved data for AFL at financial_data/AFL.csv
Saved data for A at financial_data/A.csv
Saved data for APD at financial_data/APD.csv
Saved data for ABNB at financial_data/ABNB.csv
Saved data for AKAM at financial_data/AKAM.csv
Saved data for ALB at financial_data/ALB.csv
Saved data for ARE at financial_data/ARE.csv
Saved data for ALGN at financial_data/ALGN.csv
Saved data for ALLE at financial_data/ALLE.csv
Saved data for LNT at financial_data/LNT.csv
Saved data for ALL at financial_data/ALL.csv
Saved data for GOOGL at financial_data/GOOGL.csv
Saved data for GOOG at financial_data/GOOG.csv
Saved data for MO at financial_data/MO.csv
Saved data for AMZN at financial_data/AMZN.csv
Saved data for AMCR at financial_data/AMCR.csv
Saved data for AMTM at financial_data/AMTM.csv
Saved data for AEE at financial_data/AEE.csv
Saved data for AEP at financial_data/AEP.csv
Saved data for AXP at financial_data/AXP.csv
Saved data for AIG at financial_data/AIG.csv
Saved data for AMT at financial_data/AMT.csv
Saved data for AWK at financial_data/AWK.csv
Saved data for AMP at financial_data/AMP.csv
Saved data for AME at financial_data/AME.csv
Saved data for AMGN at financial_data/AMGN.csv
Saved data for APH at financial_data/APH.csv
Saved data for ADI at financial_data/ADI.csv
Saved data for ANSS at financial_data/ANSS.csv
Saved data for AON at financial_data/AON.csv
Saved data for APA at financial_data/APA.csv
Saved data for AAPL at financial_data/AAPL.csv
Saved data for AMAT at financial_data/AMAT.csv
Saved data for APTV at financial_data/APTV.csv
Saved data for ACGL at financial_data/ACGL.csv
Saved data for ADM at financial_data/ADM.csv
Saved data for ANET at financial_data/ANET.csv
Saved data for AJG at financial_data/AJG.csv
Saved data for AIZ at financial_data/AIZ.csv
Saved data for T at financial_data/T.csv
Saved data for ATO at financial_data/ATO.csv
Saved data for ADSK at financial_data/ADSK.csv
Saved data for ADP at financial_data/ADP.csv
Saved data for AZO at financial_data/AZO.csv
Saved data for AVB at financial_data/AVB.csv
Saved data for AVY at financial_data/AVY.csv
Saved data for AXON at financial_data/AXON.csv
```

```
Saved data for BKR at financial_data/BKR.csv
Saved data for BALL at financial_data/BALL.csv
Saved data for BAC at financial_data/BAC.csv
Saved data for BAX at financial_data/BAX.csv
Saved data for BDX at financial_data/BDX.csv
Saved data for BRK-B at financial_data/BRK-B.csv
Saved data for BBY at financial_data/BBY.csv
Saved data for TECH at financial_data/TECH.csv
Saved data for BIIB at financial_data/BIIB.csv
Saved data for BLK at financial_data/BLK.csv
Saved data for BX at financial_data/BX.csv
Saved data for BK at financial_data/BK.csv
Saved data for BA at financial_data/BA.csv
Saved data for BKNG at financial_data/BKNG.csv
Saved data for BWA at financial_data/BWA.csv
Saved data for BSX at financial_data/BSX.csv
Saved data for BMY at financial_data/BMY.csv
Saved data for AVGO at financial_data/AVGO.csv
Saved data for BR at financial_data/BR.csv
Saved data for BRO at financial_data/BRO.csv
Saved data for BF-B at financial_data/BF-B.csv
Saved data for BLDR at financial_data/BLDR.csv
Saved data for BG at financial_data/BG.csv
Saved data for BXP at financial_data/BXP.csv
Saved data for CHRW at financial_data/CHRW.csv
Saved data for CDNS at financial_data/CDNS.csv
Saved data for CZR at financial_data/CZR.csv
Saved data for CPT at financial_data/CPT.csv
Saved data for CPB at financial_data/CPB.csv
Saved data for COF at financial_data/COF.csv
Saved data for CAH at financial_data/CAH.csv
Saved data for KMX at financial_data/KMX.csv
Saved data for CCL at financial_data/CCL.csv
Saved data for CARR at financial_data/CARR.csv
Saved data for CTLT at financial_data/CTLT.csv
Saved data for CAT at financial_data/CAT.csv
Saved data for CBOE at financial_data/CBOE.csv
Saved data for CBRE at financial_data/CBRE.csv
Saved data for CDW at financial_data/CDW.csv
Saved data for CE at financial_data/CE.csv
Saved data for COR at financial_data/COR.csv
Saved data for CNC at financial_data/CNC.csv
Saved data for CNP at financial_data/CNP.csv
Saved data for CF at financial_data/CF.csv
Saved data for CRL at financial_data/CRL.csv
Saved data for SCHW at financial_data/SCHW.csv
Saved data for CHTR at financial_data/CHTR.csv
Saved data for CVX at financial_data/CVX.csv
```

```
Saved data for CMG at financial_data/CMG.csv
Saved data for CB at financial_data/CB.csv
Saved data for CHD at financial_data/CHD.csv
Saved data for CI at financial_data/CI.csv
Saved data for CINF at financial_data/CINF.csv
Saved data for CTAS at financial_data/CTAS.csv
Saved data for CSCO at financial_data/CSCO.csv
Saved data for C at financial_data/C.csv
Saved data for CFG at financial_data/CFG.csv
Saved data for CLX at financial_data/CLX.csv
Saved data for CME at financial_data/CME.csv
Saved data for CMS at financial_data/CMS.csv
Saved data for KO at financial_data/KO.csv
Saved data for CTSH at financial_data/CTSH.csv
Saved data for CL at financial_data/CL.csv
Saved data for CMCSA at financial_data/CMCSA.csv
Saved data for CAG at financial_data/CAG.csv
Saved data for COP at financial_data/COP.csv
Saved data for ED at financial_data/ED.csv
Saved data for STZ at financial_data/STZ.csv
Saved data for CEG at financial_data/CEG.csv
Saved data for COO at financial_data/COO.csv
Saved data for CPRT at financial_data/CPRT.csv
Saved data for GLW at financial_data/GLW.csv
Saved data for CPAY at financial_data/CPAY.csv
Saved data for CTVA at financial_data/CTVA.csv
Saved data for CSGP at financial_data/CSGP.csv
Saved data for COST at financial_data/COST.csv
Saved data for CTRA at financial_data/CTRA.csv
Saved data for CRWD at financial_data/CRWD.csv
Saved data for CCI at financial_data/CCI.csv
Saved data for CSX at financial_data/CSX.csv
Saved data for CMI at financial_data/CMI.csv
Saved data for CVS at financial_data/CVS.csv
Saved data for DHR at financial_data/DHR.csv
Saved data for DRI at financial_data/DRI.csv
Saved data for DVA at financial_data/DVA.csv
Saved data for DAY at financial_data/DAY.csv
Saved data for DECK at financial_data/DECK.csv
Saved data for DE at financial_data/DE.csv
Saved data for DELL at financial_data/DELL.csv
Saved data for DAL at financial_data/DAL.csv
Saved data for DVN at financial_data/DVN.csv
Saved data for DXCM at financial_data/DXCM.csv
Saved data for FANG at financial_data/FANG.csv
Saved data for DLR at financial_data/DLR.csv
Saved data for DFS at financial_data/DFS.csv
Saved data for DG at financial_data/DG.csv
```

```
Saved data for DLTR at financial_data/DLTR.csv
Saved data for D at financial_data/D.csv
Saved data for DPZ at financial_data/DPZ.csv
Saved data for DOV at financial_data/DOV.csv
Saved data for DOW at financial_data/DOW.csv
Saved data for DHI at financial_data/DHI.csv
Saved data for DTE at financial_data/DTE.csv
Saved data for DUK at financial_data/DUK.csv
Saved data for DD at financial_data/DD.csv
Saved data for EMN at financial_data/EMN.csv
Saved data for ETN at financial_data/ETN.csv
Saved data for EBAY at financial_data/EBAY.csv
Saved data for ECL at financial_data/ECL.csv
Saved data for EIX at financial_data/EIX.csv
Saved data for EW at financial_data/EW.csv
Saved data for EA at financial_data/EA.csv
Saved data for ELV at financial_data/ELV.csv
Saved data for EMR at financial_data/EMR.csv
Saved data for ENPH at financial_data/ENPH.csv
Saved data for ETR at financial_data/ETR.csv
Saved data for EOG at financial_data/EOG.csv
Saved data for EPAM at financial_data/EPAM.csv
Saved data for EQT at financial_data/EQT.csv
Saved data for EFX at financial_data/EFX.csv
Saved data for EQIX at financial_data/EQIX.csv
Saved data for EQR at financial_data/EQR.csv
Saved data for ERIE at financial_data/ERIE.csv
Saved data for ESS at financial_data/ESS.csv
Saved data for EL at financial_data/EL.csv
Saved data for EG at financial_data/EG.csv
Saved data for EVRG at financial_data/EVRG.csv
Saved data for ES at financial_data/ES.csv
Saved data for EXC at financial_data/EXC.csv
Saved data for EXPE at financial_data/EXPE.csv
Saved data for EXPD at financial_data/EXPD.csv
Saved data for EXR at financial_data/EXR.csv
Saved data for XOM at financial_data/XOM.csv
Saved data for FFIV at financial_data/FFIV.csv
Saved data for FDS at financial_data/FDS.csv
Saved data for FICO at financial_data/FICO.csv
Saved data for FAST at financial_data/FAST.csv
Saved data for FRT at financial_data/FRT.csv
Saved data for FDX at financial_data/FDX.csv
Saved data for FIS at financial_data/FIS.csv
Saved data for FITB at financial_data/FITB.csv
Saved data for FSLR at financial_data/FSLR.csv
Saved data for FE at financial_data/FE.csv
Saved data for FI at financial_data/FI.csv
```

```
Saved data for FMC at financial_data/FMC.csv
Saved data for F at financial_data/F.csv
Saved data for FTNT at financial_data/FTNT.csv
Saved data for FTV at financial_data/FTV.csv
Saved data for FOXA at financial_data/FOXA.csv
Saved data for FOX at financial_data/FOX.csv
Saved data for BEN at financial_data/BEN.csv
Saved data for FCX at financial_data/FCX.csv
Saved data for GRMN at financial_data/GRMN.csv
Saved data for IT at financial_data/IT.csv
Saved data for GE at financial_data/GE.csv
Saved data for GEHC at financial_data/GEHC.csv
Saved data for GEV at financial_data/GEV.csv
Saved data for GEN at financial_data/GEN.csv
Saved data for GNRC at financial_data/GNRC.csv
Saved data for GD at financial_data/GD.csv
Saved data for GIS at financial_data/GIS.csv
Saved data for GM at financial_data/GM.csv
Saved data for GPC at financial_data/GPC.csv
Saved data for GILD at financial_data/GILD.csv
Saved data for GPN at financial_data/GPN.csv
Saved data for GL at financial_data/GL.csv
Saved data for GDDY at financial_data/GDDY.csv
Saved data for GS at financial_data/GS.csv
Saved data for HAL at financial_data/HAL.csv
Saved data for HIG at financial_data/HIG.csv
Saved data for HAS at financial_data/HAS.csv
Saved data for HCA at financial_data/HCA.csv
Saved data for DOC at financial_data/DOC.csv
Saved data for HSIC at financial_data/HSIC.csv
Saved data for HSY at financial_data/HSY.csv
Saved data for HES at financial_data/HES.csv
Saved data for HPE at financial_data/HPE.csv
Saved data for HLT at financial_data/HLT.csv
Saved data for HOLX at financial_data/HOLX.csv
Saved data for HD at financial_data/HD.csv
Saved data for HON at financial_data/HON.csv
Saved data for HRL at financial_data/HRL.csv
Saved data for HST at financial_data/HST.csv
Saved data for HWM at financial_data/HWM.csv
Saved data for HPQ at financial_data/HPQ.csv
Saved data for HUBB at financial_data/HUBB.csv
Saved data for HUM at financial_data/HUM.csv
Saved data for HBAN at financial_data/HBAN.csv
Saved data for HII at financial_data/HII.csv
Saved data for IBM at financial_data/IBM.csv
Saved data for IEX at financial_data/IEX.csv
Saved data for IDXX at financial_data/IDXX.csv
```

```
Saved data for ITW at financial_data/ITW.csv
Saved data for INCY at financial_data/INCY.csv
Saved data for IR at financial_data/IR.csv
Saved data for PODD at financial_data/PODD.csv
Saved data for INTC at financial_data/INTC.csv
Saved data for ICE at financial_data/ICE.csv
Saved data for IFF at financial_data/IFF.csv
Saved data for IP at financial_data/IP.csv
Saved data for IPG at financial_data/IPG.csv
Saved data for INTU at financial_data/INTU.csv
Saved data for ISRG at financial_data/ISRG.csv
Saved data for IVZ at financial_data/IVZ.csv
Saved data for INVH at financial_data/INVH.csv
Saved data for IQV at financial_data/IQV.csv
Saved data for IRM at financial_data/IRM.csv
Saved data for JBHT at financial_data/JBHT.csv
Saved data for JBL at financial_data/JBL.csv
Saved data for JKHY at financial_data/JKHY.csv
Saved data for J at financial_data/J.csv
Saved data for JNJ at financial_data/JNJ.csv
Saved data for JCI at financial_data/JCI.csv
Saved data for JPM at financial_data/JPM.csv
Saved data for JNPR at financial_data/JNPR.csv
Saved data for K at financial_data/K.csv
Saved data for KVUE at financial_data/KVUE.csv
Saved data for KDP at financial_data/KDP.csv
Saved data for KEY at financial_data/KEY.csv
Saved data for KEYS at financial_data/KEYS.csv
Saved data for KMB at financial_data/KMB.csv
Saved data for KIM at financial_data/KIM.csv
Saved data for KMI at financial_data/KMI.csv
Saved data for KKR at financial_data/KKR.csv
Saved data for KLAC at financial_data/KLAC.csv
Saved data for KHC at financial_data/KHC.csv
Saved data for KR at financial_data/KR.csv
Saved data for LHX at financial_data/LHX.csv
Saved data for LH at financial_data/LH.csv
Saved data for LRCX at financial_data/LRCX.csv
Saved data for LW at financial_data/LW.csv
Saved data for LVS at financial_data/LVS.csv
Saved data for LDOS at financial_data/LDOS.csv
Saved data for LEN at financial_data/LEN.csv
Saved data for LLY at financial_data/LLY.csv
Saved data for LIN at financial_data/LIN.csv
Saved data for LYV at financial_data/LYV.csv
Saved data for LKQ at financial_data/LKQ.csv
Saved data for LMT at financial_data/LMT.csv
Saved data for L at financial_data/L.csv
```

```
Saved data for LOW at financial_data/LOW.csv
Saved data for LULU at financial_data/LULU.csv
Saved data for LYB at financial_data/LYB.csv
Saved data for MTB at financial_data/MTB.csv
Saved data for MPC at financial_data/MPC.csv
Saved data for MKTX at financial_data/MKTX.csv
Saved data for MAR at financial_data/MAR.csv
Saved data for MMC at financial_data/MMC.csv
Saved data for MLM at financial_data/MLM.csv
Saved data for MAS at financial_data/MAS.csv
Saved data for MA at financial_data/MA.csv
Saved data for MTCH at financial_data/MTCH.csv
Saved data for MKC at financial_data/MKC.csv
Saved data for MCD at financial_data/MCD.csv
Saved data for MCK at financial_data/MCK.csv
Saved data for MDT at financial_data/MDT.csv
Saved data for MRK at financial_data/MRK.csv
Saved data for META at financial_data/META.csv
Saved data for MET at financial_data/MET.csv
Saved data for MTD at financial_data/MTD.csv
Saved data for MGM at financial_data/MGM.csv
Saved data for MCHP at financial_data/MCHP.csv
Saved data for MU at financial_data/MU.csv
Saved data for MSFT at financial_data/MSFT.csv
Saved data for MAA at financial_data/MAA.csv
Saved data for MRNA at financial_data/MRNA.csv
Saved data for MHK at financial_data/MHK.csv
Saved data for MOH at financial_data/MOH.csv
Saved data for TAP at financial_data/TAP.csv
Saved data for MDLZ at financial_data/MDLZ.csv
Saved data for MPWR at financial_data/MPWR.csv
Saved data for MNST at financial_data/MNST.csv
Saved data for MCO at financial_data/MCO.csv
Saved data for MS at financial_data/MS.csv
Saved data for MOS at financial_data/MOS.csv
Saved data for MSI at financial_data/MSI.csv
Saved data for MSCI at financial_data/MSCI.csv
Saved data for NDAQ at financial_data/NDAQ.csv
Saved data for NTAP at financial_data/NTAP.csv
Saved data for NFLX at financial_data/NFLX.csv
Saved data for NEM at financial_data/NEM.csv
Saved data for NWSA at financial_data/NWSA.csv
Saved data for NWS at financial_data/NWS.csv
Saved data for NEE at financial_data/NEE.csv
Saved data for NKE at financial_data/NKE.csv
Saved data for NI at financial_data/NI.csv
Saved data for NDSN at financial_data/NDSN.csv
Saved data for NSC at financial_data/NSC.csv
```

```
Saved data for NTRS at financial_data/NTRS.csv
Saved data for NOC at financial_data/NOC.csv
Saved data for NCLH at financial_data/NCLH.csv
Saved data for NRG at financial_data/NRG.csv
Saved data for NUE at financial_data/NUE.csv
Saved data for NVDA at financial_data/NVDA.csv
Saved data for NVR at financial_data/NVR.csv
Saved data for NXPI at financial_data/NXPI.csv
Saved data for ORLY at financial_data/ORLY.csv
Saved data for OXY at financial_data/OXY.csv
Saved data for ODFL at financial_data/ODFL.csv
Saved data for OMC at financial_data/OMC.csv
Saved data for ON at financial_data/ON.csv
Saved data for OKE at financial_data/OKE.csv
Saved data for ORCL at financial_data/ORCL.csv
Saved data for OTIS at financial_data/OTIS.csv
Saved data for PCAR at financial_data/PCAR.csv
Saved data for PKG at financial_data/PKG.csv
Saved data for PLTR at financial_data/PLTR.csv
Saved data for PANW at financial_data/PANW.csv
Saved data for PARA at financial_data/PARA.csv
Saved data for PH at financial_data/PH.csv
Saved data for PAYX at financial_data/PAYX.csv
Saved data for PAYC at financial_data/PAYC.csv
Saved data for PYPL at financial_data/PYPL.csv
Saved data for PNR at financial_data/PNR.csv
Saved data for PEP at financial_data/PEP.csv
Saved data for PFE at financial_data/PFE.csv
Saved data for PCG at financial_data/PCG.csv
Saved data for PM at financial_data/PM.csv
Saved data for PSX at financial_data/PSX.csv
Saved data for PNW at financial_data/PNW.csv
Saved data for PNC at financial_data/PNC.csv
Saved data for POOL at financial_data/POOL.csv
Saved data for PPG at financial_data/PPG.csv
Saved data for PPL at financial_data/PPL.csv
Saved data for PFG at financial_data/PFG.csv
Saved data for PG at financial_data/PG.csv
Saved data for PGR at financial_data/PGR.csv
Saved data for PLD at financial_data/PLD.csv
Saved data for PRU at financial_data/PRU.csv
Saved data for PEG at financial_data/PEG.csv
Saved data for PTC at financial_data/PTC.csv
Saved data for PSA at financial_data/PSA.csv
Saved data for PHM at financial_data/PHM.csv
Saved data for QRVO at financial_data/QRVO.csv
Saved data for PWR at financial_data/PWR.csv
Saved data for QCOM at financial_data/QCOM.csv
```

```
Saved data for DGX at financial_data/DGX.csv
Saved data for RL at financial_data/RL.csv
Saved data for RJF at financial_data/RJF.csv
Saved data for RTX at financial_data/RTX.csv
Saved data for O at financial_data/O.csv
Saved data for REG at financial_data/REG.csv
Saved data for REGN at financial_data/REGN.csv
Saved data for RF at financial_data/RF.csv
Saved data for RSG at financial_data/RSG.csv
Saved data for RMD at financial_data/RMD.csv
Saved data for RVTY at financial_data/RVTY.csv
Saved data for ROK at financial_data/ROK.csv
Saved data for ROL at financial_data/ROL.csv
Saved data for ROP at financial_data/ROP.csv
Saved data for ROST at financial_data/ROST.csv
Saved data for RCL at financial_data/RCL.csv
Saved data for SPGI at financial_data/SPGI.csv
Saved data for CRM at financial_data/CRM.csv
Saved data for SBAC at financial_data/SBAC.csv
Saved data for SLB at financial_data/SLB.csv
Saved data for STX at financial_data/STX.csv
Saved data for SRE at financial_data/SRE.csv
Saved data for NOW at financial_data/NOW.csv
Saved data for SHW at financial_data/SHW.csv
Saved data for SPG at financial_data/SPG.csv
Saved data for SWKS at financial_data/SWKS.csv
Saved data for SJM at financial_data/SJM.csv
Saved data for SW at financial_data/SW.csv
Saved data for SNA at financial_data/SNA.csv
Saved data for SOLV at financial_data/SOLV.csv
Saved data for SO at financial_data/SO.csv
Saved data for LUV at financial_data/LUV.csv
Saved data for SWK at financial_data/SWK.csv
Saved data for SBUX at financial_data/SBUX.csv
Saved data for STT at financial_data/STT.csv
Saved data for STLD at financial_data/STLD.csv
Saved data for STE at financial_data/STE.csv
Saved data for SYK at financial_data/SYK.csv
Saved data for SMCI at financial_data/SMCI.csv
Saved data for SYF at financial_data/SYF.csv
Saved data for SNPS at financial_data/SNPS.csv
Saved data for SYY at financial_data/SYY.csv
Saved data for TMUS at financial_data/TMUS.csv
Saved data for TROW at financial_data/TROW.csv
Saved data for TTWO at financial_data/TTWO.csv
Saved data for TPR at financial_data/TPR.csv
Saved data for TRGP at financial_data/TRGP.csv
Saved data for TGT at financial_data/TGT.csv
```

```
Saved data for TEL at financial_data/TEL.csv
Saved data for TDY at financial_data/TDY.csv
Saved data for TFX at financial_data/TFX.csv
Saved data for TER at financial_data/TER.csv
Saved data for TSLA at financial_data/TSLA.csv
Saved data for TXN at financial_data/TXN.csv
Saved data for TPL at financial_data/TPL.csv
Saved data for TXT at financial_data/TXT.csv
Saved data for TMO at financial_data/TMO.csv
Saved data for TJX at financial_data/TJX.csv
Saved data for TSCO at financial_data/TSCO.csv
Saved data for TT at financial_data/TT.csv
Saved data for TDG at financial_data/TDG.csv
Saved data for TRV at financial_data/TRV.csv
Saved data for TRMB at financial_data/TRMB.csv
Saved data for TFC at financial_data/TFC.csv
Saved data for TYL at financial_data/TYL.csv
Saved data for TSN at financial_data/TSN.csv
Saved data for USB at financial_data/USB.csv
Saved data for UBER at financial_data/UBER.csv
Saved data for UDR at financial_data/UDR.csv
Saved data for ULTA at financial_data/ULTA.csv
Saved data for UNP at financial_data/UNP.csv
Saved data for UAL at financial_data/UAL.csv
Saved data for UPS at financial_data/UPS.csv
Saved data for URI at financial_data/URI.csv
Saved data for UNH at financial_data/UNH.csv
Saved data for UHS at financial_data/UHS.csv
Saved data for VLO at financial_data/VLO.csv
Saved data for VTR at financial_data/VTR.csv
Saved data for VLTO at financial_data/VLTO.csv
Saved data for VRSN at financial_data/VRSN.csv
Saved data for VRSK at financial_data/VRSK.csv
Saved data for VZ at financial_data/VZ.csv
Saved data for VRTX at financial_data/VRTX.csv
Saved data for VTRS at financial_data/VTRS.csv
Saved data for VICI at financial_data/VICI.csv
Saved data for V at financial_data/V.csv
Saved data for VST at financial_data/VST.csv
Saved data for VMC at financial_data/VMC.csv
Saved data for WRB at financial_data/WRB.csv
Saved data for GWW at financial_data/GWW.csv
Saved data for WAB at financial_data/WAB.csv
Saved data for WBA at financial_data/WBA.csv
Saved data for WMT at financial_data/WMT.csv
Saved data for DIS at financial_data/DIS.csv
Saved data for WBD at financial_data/WBD.csv
Saved data for WM at financial_data/WM.csv
```

```
Saved data for WAT at financial_data/WAT.csv
Saved data for WEC at financial_data/WEC.csv
Saved data for WFC at financial_data/WFC.csv
Saved data for WELL at financial_data/WELL.csv
Saved data for WST at financial_data/WST.csv
Saved data for WDC at financial_data/WDC.csv
Saved data for WY at financial_data/WY.csv
Saved data for WMB at financial_data/WMB.csv
Saved data for WTW at financial_data/WTW.csv
Saved data for WYNN at financial_data/WYNN.csv
Saved data for XEL at financial_data/XEL.csv
Saved data for XYL at financial_data/XYL.csv
Saved data for YUM at financial_data/YUM.csv
Saved data for ZBRA at financial_data/ZBRA.csv
Saved data for ZBH at financial_data/ZBH.csv
Saved data for ZTS at financial_data/ZTS.csv
```

#Data Cleaning and Preprocessing

[5]:
```
import os
import pandas as pd

def load_and_clean_data(folder="financial_data"):
    all_data = {}
    files = os.listdir(folder)

    for file in files:
        if file.endswith(".csv"):
            file_path = os.path.join(folder, file)
            ticker = file.replace(".csv", "")

            try:
                # Skip first two rows, set proper headers
                columns = ['Date', 'Adj Close', 'Close', 'High', 'Low', 'Open',␣
 ↪'Volume']
                df = pd.read_csv(file_path, skiprows=2, names=columns)

                # Drop any rows with missing data
                df.dropna(inplace=True)

                # Ensure correct data types
                df['Date'] = pd.to_datetime(df['Date'])
                df[['Adj Close', 'Close', 'High', 'Low', 'Open', 'Volume']] =␣
 ↪df[
                    ['Adj Close', 'Close', 'High', 'Low', 'Open', 'Volume']
                ].apply(pd.to_numeric, errors='coerce')

                all_data[ticker] = df
```

```
                print(f"Loaded and cleaned data for {ticker}")
            except Exception as e:
                print(f"Failed to process {file}: {e}")

    return all_data

# Run the loading and cleaning process
financial_data_cleaned = load_and_clean_data()
```

```
Loaded and cleaned data for MS
Loaded and cleaned data for CMS
Loaded and cleaned data for GM
Loaded and cleaned data for BSX
Loaded and cleaned data for AON
Loaded and cleaned data for ETN
Loaded and cleaned data for ITW
Loaded and cleaned data for NRG
Loaded and cleaned data for IEX
Loaded and cleaned data for ABNB
Loaded and cleaned data for VTR
Loaded and cleaned data for TEL
Loaded and cleaned data for NVDA
Loaded and cleaned data for HST
Loaded and cleaned data for AAPL
Loaded and cleaned data for F
Loaded and cleaned data for DELL
Loaded and cleaned data for PCAR
Loaded and cleaned data for CTLT
Loaded and cleaned data for GOOGL
Loaded and cleaned data for GS
Loaded and cleaned data for JNJ
Loaded and cleaned data for TECH
Loaded and cleaned data for CPRT
Loaded and cleaned data for COO
Loaded and cleaned data for APA
Loaded and cleaned data for DLTR
Loaded and cleaned data for BXP
Loaded and cleaned data for RCL
Loaded and cleaned data for MCD
Loaded and cleaned data for FE
Loaded and cleaned data for MCK
Loaded and cleaned data for HUM
Loaded and cleaned data for SYK
Loaded and cleaned data for BWA
Loaded and cleaned data for TER
Loaded and cleaned data for LYV
Loaded and cleaned data for NFLX
```

```
Loaded and cleaned data for MLM
Loaded and cleaned data for UHS
Loaded and cleaned data for UNP
Loaded and cleaned data for SJM
Loaded and cleaned data for JCI
Loaded and cleaned data for AMCR
Loaded and cleaned data for BK
Loaded and cleaned data for LW
Loaded and cleaned data for SYY
Loaded and cleaned data for RJF
Loaded and cleaned data for KKR
Loaded and cleaned data for NWS
Loaded and cleaned data for MKC
Loaded and cleaned data for TRGP
Loaded and cleaned data for ON
Loaded and cleaned data for MTCH
Loaded and cleaned data for LLY
Loaded and cleaned data for ORLY
Loaded and cleaned data for CBRE
Loaded and cleaned data for FANG
Loaded and cleaned data for K
Loaded and cleaned data for CPAY
Loaded and cleaned data for MDT
Loaded and cleaned data for CE
Loaded and cleaned data for NVR
Loaded and cleaned data for SRE
Loaded and cleaned data for APD
Loaded and cleaned data for PWR
Loaded and cleaned data for MDLZ
Loaded and cleaned data for ISRG
Loaded and cleaned data for MAS
Loaded and cleaned data for FICO
Loaded and cleaned data for PAYC
Loaded and cleaned data for CTRA
Loaded and cleaned data for MMM
Loaded and cleaned data for ROP
Loaded and cleaned data for DOC
Loaded and cleaned data for MPWR
Loaded and cleaned data for GWW
Loaded and cleaned data for MOH
Loaded and cleaned data for FI
Loaded and cleaned data for ROL
Loaded and cleaned data for CBOE
Loaded and cleaned data for GL
Loaded and cleaned data for MHK
Loaded and cleaned data for HPE
Loaded and cleaned data for JPM
Loaded and cleaned data for TFX
```

```
Loaded and cleaned data for NCLH
Loaded and cleaned data for MSCI
Loaded and cleaned data for CPT
Loaded and cleaned data for IRM
Loaded and cleaned data for MAR
Loaded and cleaned data for IVZ
Loaded and cleaned data for EXC
Loaded and cleaned data for QCOM
Loaded and cleaned data for WBA
Loaded and cleaned data for NOW
Loaded and cleaned data for ADM
Loaded and cleaned data for WRB
Loaded and cleaned data for VMC
Loaded and cleaned data for SLB
Loaded and cleaned data for KVUE
Loaded and cleaned data for NI
Loaded and cleaned data for ROST
Loaded and cleaned data for WY
Loaded and cleaned data for SWKS
Loaded and cleaned data for CTVA
Loaded and cleaned data for NDAQ
Loaded and cleaned data for DVA
Loaded and cleaned data for EQIX
Loaded and cleaned data for AZO
Loaded and cleaned data for CMI
Loaded and cleaned data for DHR
Loaded and cleaned data for KR
Loaded and cleaned data for NTAP
Loaded and cleaned data for FOX
Loaded and cleaned data for ABBV
Loaded and cleaned data for FFIV
Loaded and cleaned data for RMD
Loaded and cleaned data for EG
Loaded and cleaned data for EA
Loaded and cleaned data for TMO
Loaded and cleaned data for MA
Loaded and cleaned data for UDR
Loaded and cleaned data for BF-B
Loaded and cleaned data for ALL
Loaded and cleaned data for CLX
Loaded and cleaned data for AME
Loaded and cleaned data for HWM
Loaded and cleaned data for A
Loaded and cleaned data for TGT
Loaded and cleaned data for WYNN
Loaded and cleaned data for BIIB
Loaded and cleaned data for OXY
Loaded and cleaned data for EIX
```

```
Loaded and cleaned data for GPN
Loaded and cleaned data for ANSS
Loaded and cleaned data for ADBE
Loaded and cleaned data for PANW
Loaded and cleaned data for HOLX
Loaded and cleaned data for BKR
Loaded and cleaned data for HLT
Loaded and cleaned data for WAT
Loaded and cleaned data for COF
Loaded and cleaned data for EPAM
Loaded and cleaned data for DPZ
Loaded and cleaned data for DAL
Loaded and cleaned data for MCHP
Loaded and cleaned data for LRCX
Loaded and cleaned data for WM
Loaded and cleaned data for FRT
Loaded and cleaned data for GE
Loaded and cleaned data for ACN
Loaded and cleaned data for DFS
Loaded and cleaned data for NKE
Loaded and cleaned data for CB
Loaded and cleaned data for INTC
Loaded and cleaned data for STZ
Loaded and cleaned data for CRM
Loaded and cleaned data for KO
Loaded and cleaned data for AMP
Loaded and cleaned data for T
Loaded and cleaned data for HBAN
Loaded and cleaned data for MKTX
Loaded and cleaned data for D
Loaded and cleaned data for RVTY
Loaded and cleaned data for ULTA
Loaded and cleaned data for L
Loaded and cleaned data for STLD
Loaded and cleaned data for LUV
Loaded and cleaned data for VLTO
Loaded and cleaned data for VICI
Loaded and cleaned data for TRV
Loaded and cleaned data for HIG
Loaded and cleaned data for BKNG
Loaded and cleaned data for CRWD
Loaded and cleaned data for IFF
Loaded and cleaned data for HUBB
Loaded and cleaned data for ZTS
Loaded and cleaned data for FDX
Loaded and cleaned data for TAP
Loaded and cleaned data for CMCSA
Loaded and cleaned data for TDY
```

```
Loaded and cleaned data for WST
Loaded and cleaned data for INTU
Loaded and cleaned data for MSI
Loaded and cleaned data for LEN
Loaded and cleaned data for CTAS
Loaded and cleaned data for LKQ
Loaded and cleaned data for COP
Loaded and cleaned data for AVB
Loaded and cleaned data for DECK
Loaded and cleaned data for AIG
Loaded and cleaned data for AOS
Loaded and cleaned data for AMTM
Loaded and cleaned data for PPG
Loaded and cleaned data for CAG
Loaded and cleaned data for DG
Loaded and cleaned data for BX
Loaded and cleaned data for CDNS
Loaded and cleaned data for USB
Loaded and cleaned data for MU
Loaded and cleaned data for TRMB
Loaded and cleaned data for AMZN
Loaded and cleaned data for EQR
Loaded and cleaned data for TSN
Loaded and cleaned data for HD
Loaded and cleaned data for RL
Loaded and cleaned data for GPC
Loaded and cleaned data for CEG
Loaded and cleaned data for CHRW
Loaded and cleaned data for PODD
Loaded and cleaned data for KEY
Loaded and cleaned data for HAS
Loaded and cleaned data for CHTR
Loaded and cleaned data for LULU
Loaded and cleaned data for DXCM
Loaded and cleaned data for TPL
Loaded and cleaned data for LHX
Loaded and cleaned data for C
Loaded and cleaned data for AWK
Loaded and cleaned data for PEG
Loaded and cleaned data for BR
Loaded and cleaned data for SBUX
Loaded and cleaned data for KEYS
Loaded and cleaned data for WBD
Loaded and cleaned data for DRI
Loaded and cleaned data for IQV
Loaded and cleaned data for VTRS
Loaded and cleaned data for ANET
Loaded and cleaned data for EMN
```

```
Loaded and cleaned data for GNRC
Loaded and cleaned data for MSFT
Loaded and cleaned data for VLO
Loaded and cleaned data for J
Loaded and cleaned data for DGX
Loaded and cleaned data for CTSH
Loaded and cleaned data for BALL
Loaded and cleaned data for WTW
Loaded and cleaned data for ADI
Loaded and cleaned data for GOOG
Loaded and cleaned data for NEE
Loaded and cleaned data for COR
Loaded and cleaned data for BRK-B
Loaded and cleaned data for DAY
Loaded and cleaned data for HAL
Loaded and cleaned data for O
Loaded and cleaned data for DIS
Loaded and cleaned data for FIS
Loaded and cleaned data for AMT
Loaded and cleaned data for YUM
Loaded and cleaned data for BEN
Loaded and cleaned data for CI
Loaded and cleaned data for XYL
Loaded and cleaned data for AES
Loaded and cleaned data for FITB
Loaded and cleaned data for VST
Loaded and cleaned data for PLTR
Loaded and cleaned data for MGM
Loaded and cleaned data for STE
Loaded and cleaned data for TYL
Loaded and cleaned data for ELV
Loaded and cleaned data for ZBRA
Loaded and cleaned data for ADSK
Loaded and cleaned data for PEP
Loaded and cleaned data for FDS
Loaded and cleaned data for AJG
Loaded and cleaned data for AMAT
Loaded and cleaned data for WFC
Loaded and cleaned data for AMGN
Loaded and cleaned data for TPR
Loaded and cleaned data for SBAC
Loaded and cleaned data for ES
Loaded and cleaned data for MTB
Loaded and cleaned data for DHI
Loaded and cleaned data for PG
Loaded and cleaned data for EW
Loaded and cleaned data for DOW
Loaded and cleaned data for CVS
```

```
Loaded and cleaned data for CRL
Loaded and cleaned data for ARE
Loaded and cleaned data for DD
Loaded and cleaned data for FSLR
Loaded and cleaned data for DOV
Loaded and cleaned data for CCL
Loaded and cleaned data for GD
Loaded and cleaned data for CAT
Loaded and cleaned data for MCO
Loaded and cleaned data for DTE
Loaded and cleaned data for CZR
Loaded and cleaned data for IP
Loaded and cleaned data for PHM
Loaded and cleaned data for BMY
Loaded and cleaned data for NSC
Loaded and cleaned data for WMT
Loaded and cleaned data for DE
Loaded and cleaned data for CAH
Loaded and cleaned data for NOC
Loaded and cleaned data for REGN
Loaded and cleaned data for MPC
Loaded and cleaned data for SWK
Loaded and cleaned data for JNPR
Loaded and cleaned data for TXN
Loaded and cleaned data for LH
Loaded and cleaned data for SHW
Loaded and cleaned data for CARR
Loaded and cleaned data for WDC
Loaded and cleaned data for PH
Loaded and cleaned data for SYF
Loaded and cleaned data for EVRG
Loaded and cleaned data for BLK
Loaded and cleaned data for DLR
Loaded and cleaned data for UBER
Loaded and cleaned data for ATO
Loaded and cleaned data for PNW
Loaded and cleaned data for TSLA
Loaded and cleaned data for OMC
Loaded and cleaned data for HRL
Loaded and cleaned data for BBY
Loaded and cleaned data for KMX
Loaded and cleaned data for CMG
Loaded and cleaned data for XOM
Loaded and cleaned data for TDG
Loaded and cleaned data for FMC
Loaded and cleaned data for JBL
Loaded and cleaned data for NDSN
Loaded and cleaned data for COST
```

```
Loaded and cleaned data for HCA
Loaded and cleaned data for ODFL
Loaded and cleaned data for RF
Loaded and cleaned data for IT
Loaded and cleaned data for FTNT
Loaded and cleaned data for SMCI
Loaded and cleaned data for ABT
Loaded and cleaned data for SPGI
Loaded and cleaned data for MO
Loaded and cleaned data for MRNA
Loaded and cleaned data for PTC
Loaded and cleaned data for EXPE
Loaded and cleaned data for IR
Loaded and cleaned data for ETR
Loaded and cleaned data for CSCO
Loaded and cleaned data for TROW
Loaded and cleaned data for FOXA
Loaded and cleaned data for NWSA
Loaded and cleaned data for NUE
Loaded and cleaned data for PGR
Loaded and cleaned data for LVS
Loaded and cleaned data for NXPI
Loaded and cleaned data for WAB
Loaded and cleaned data for DUK
Loaded and cleaned data for LMT
Loaded and cleaned data for AEE
Loaded and cleaned data for INVH
Loaded and cleaned data for V
Loaded and cleaned data for SW
Loaded and cleaned data for GIS
Loaded and cleaned data for ORCL
Loaded and cleaned data for GEV
Loaded and cleaned data for AIZ
Loaded and cleaned data for STX
Loaded and cleaned data for TT
Loaded and cleaned data for JBHT
Loaded and cleaned data for ED
Loaded and cleaned data for HSY
Loaded and cleaned data for GDDY
Loaded and cleaned data for TJX
Loaded and cleaned data for EL
Loaded and cleaned data for SOLV
Loaded and cleaned data for GEN
Loaded and cleaned data for PNC
Loaded and cleaned data for MTD
Loaded and cleaned data for PCG
Loaded and cleaned data for PLD
Loaded and cleaned data for REG
```

```
Loaded and cleaned data for UAL
Loaded and cleaned data for SPG
Loaded and cleaned data for NTRS
Loaded and cleaned data for SO
Loaded and cleaned data for CSX
Loaded and cleaned data for EQT
Loaded and cleaned data for IBM
Loaded and cleaned data for MRK
Loaded and cleaned data for BAC
Loaded and cleaned data for QRVO
Loaded and cleaned data for EBAY
Loaded and cleaned data for PAYX
Loaded and cleaned data for JKHY
Loaded and cleaned data for AXON
Loaded and cleaned data for ALGN
Loaded and cleaned data for DVN
Loaded and cleaned data for FCX
Loaded and cleaned data for VZ
Loaded and cleaned data for GRMN
Loaded and cleaned data for TSCO
Loaded and cleaned data for PFE
Loaded and cleaned data for HES
Loaded and cleaned data for CF
Loaded and cleaned data for URI
Loaded and cleaned data for GLW
Loaded and cleaned data for TMUS
Loaded and cleaned data for PSX
Loaded and cleaned data for CCI
Loaded and cleaned data for RTX
Loaded and cleaned data for EXPD
Loaded and cleaned data for WMB
Loaded and cleaned data for HII
Loaded and cleaned data for FTV
Loaded and cleaned data for CINF
Loaded and cleaned data for PPL
Loaded and cleaned data for SNA
Loaded and cleaned data for AEP
Loaded and cleaned data for GILD
Loaded and cleaned data for AFL
Loaded and cleaned data for ALB
Loaded and cleaned data for KIM
Loaded and cleaned data for MMC
Loaded and cleaned data for META
Loaded and cleaned data for EMR
Loaded and cleaned data for AXP
Loaded and cleaned data for BLDR
Loaded and cleaned data for ADP
Loaded and cleaned data for PRU
```

```
Loaded and cleaned data for CPB
Loaded and cleaned data for OTIS
Loaded and cleaned data for BRO
Loaded and cleaned data for GEHC
Loaded and cleaned data for MET
Loaded and cleaned data for BG
Loaded and cleaned data for BAX
Loaded and cleaned data for INCY
Loaded and cleaned data for POOL
Loaded and cleaned data for AVGO
Loaded and cleaned data for XEL
Loaded and cleaned data for CDW
Loaded and cleaned data for PFG
Loaded and cleaned data for UNH
Loaded and cleaned data for TTWO
Loaded and cleaned data for LNT
Loaded and cleaned data for ACGL
Loaded and cleaned data for HPQ
Loaded and cleaned data for HSIC
Loaded and cleaned data for ROK
Loaded and cleaned data for CVX
Loaded and cleaned data for PYPL
Loaded and cleaned data for TFC
Loaded and cleaned data for CL
Loaded and cleaned data for RSG
Loaded and cleaned data for CME
Loaded and cleaned data for KMI
Loaded and cleaned data for BDX
Loaded and cleaned data for ECL
Loaded and cleaned data for IPG
Loaded and cleaned data for TXT
Loaded and cleaned data for CNP
Loaded and cleaned data for LOW
Loaded and cleaned data for AKAM
Loaded and cleaned data for PKG
Loaded and cleaned data for EFX
Loaded and cleaned data for APTV
Loaded and cleaned data for PSA
Loaded and cleaned data for STT
Loaded and cleaned data for VRSK
Loaded and cleaned data for APH
Loaded and cleaned data for ESS
Loaded and cleaned data for KLAC
Loaded and cleaned data for KDP
Loaded and cleaned data for CNC
Loaded and cleaned data for LDOS
Loaded and cleaned data for PARA
Loaded and cleaned data for HON
```

```
Loaded and cleaned data for LYB
Loaded and cleaned data for MNST
Loaded and cleaned data for EXR
Loaded and cleaned data for AVY
Loaded and cleaned data for SNPS
Loaded and cleaned data for ICE
Loaded and cleaned data for KHC
Loaded and cleaned data for VRSN
Loaded and cleaned data for SCHW
Loaded and cleaned data for ALLE
Loaded and cleaned data for ERIE
Loaded and cleaned data for WELL
Loaded and cleaned data for ENPH
Loaded and cleaned data for EOG
Loaded and cleaned data for MOS
Loaded and cleaned data for CHD
Loaded and cleaned data for WEC
Loaded and cleaned data for UPS
Loaded and cleaned data for ZBH
Loaded and cleaned data for CFG
Loaded and cleaned data for CSGP
Loaded and cleaned data for OKE
Loaded and cleaned data for FAST
Loaded and cleaned data for VRTX
Loaded and cleaned data for PM
Loaded and cleaned data for MAA
Loaded and cleaned data for IDXX
Loaded and cleaned data for BA
Loaded and cleaned data for KMB
Loaded and cleaned data for LIN
Loaded and cleaned data for AMD
Loaded and cleaned data for PNR
Loaded and cleaned data for NEM
```

[6]:
```python
def preprocess_for_embeddings(data):
    processed_data = []

    for ticker, df in data.items():
        if df.empty:
            continue

        # Calculate financial statistics
        summary = {
            "Ticker": ticker,
            "Mean_Close": df['Close'].mean(),
            "Max_Close": df['Close'].max(),
            "Min_Close": df['Close'].min(),
```

```
            "Std_Close": df['Close'].std(),
            "Return_3m": (df['Close'].iloc[-1] - df['Close'].iloc[0]) /␣
 ↪df['Close'].iloc[0],
            "Latest_Close": df['Close'].iloc[-1],
            "Latest_Volume": df['Volume'].iloc[-1],
            "Latest_Date": df['Date'].iloc[-1].strftime("%Y-%m-%d"),
        }

        processed_data.append(summary)
        print(f"Processed data for {ticker}")

    # Convert to DataFrame and save
    processed_df = pd.DataFrame(processed_data)
    processed_df.to_csv("processed_financial_data.csv", index=False)
    return processed_df

# Run preprocessing
processed_financial_data = preprocess_for_embeddings(financial_data_cleaned)
```

```
Processed data for MS
Processed data for CMS
Processed data for GM
Processed data for BSX
Processed data for AON
Processed data for ETN
Processed data for ITW
Processed data for NRG
Processed data for IEX
Processed data for ABNB
Processed data for VTR
Processed data for TEL
Processed data for NVDA
Processed data for HST
Processed data for AAPL
Processed data for F
Processed data for DELL
Processed data for PCAR
Processed data for CTLT
Processed data for GOOGL
Processed data for GS
Processed data for JNJ
Processed data for TECH
Processed data for CPRT
Processed data for COO
Processed data for APA
Processed data for DLTR
Processed data for BXP
```

```
Processed data for RCL
Processed data for MCD
Processed data for FE
Processed data for MCK
Processed data for HUM
Processed data for SYK
Processed data for BWA
Processed data for TER
Processed data for LYV
Processed data for NFLX
Processed data for MLM
Processed data for UHS
Processed data for UNP
Processed data for SJM
Processed data for JCI
Processed data for AMCR
Processed data for BK
Processed data for LW
Processed data for SYY
Processed data for RJF
Processed data for KKR
Processed data for NWS
Processed data for MKC
Processed data for TRGP
Processed data for ON
Processed data for MTCH
Processed data for LLY
Processed data for ORLY
Processed data for CBRE
Processed data for FANG
Processed data for K
Processed data for CPAY
Processed data for MDT
Processed data for CE
Processed data for NVR
Processed data for SRE
Processed data for APD
Processed data for PWR
Processed data for MDLZ
Processed data for ISRG
Processed data for MAS
Processed data for FICO
Processed data for PAYC
Processed data for CTRA
Processed data for MMM
Processed data for ROP
Processed data for DOC
Processed data for MPWR
```

```
Processed data for GWW
Processed data for MOH
Processed data for FI
Processed data for ROL
Processed data for CBOE
Processed data for GL
Processed data for MHK
Processed data for HPE
Processed data for JPM
Processed data for TFX
Processed data for NCLH
Processed data for MSCI
Processed data for CPT
Processed data for IRM
Processed data for MAR
Processed data for IVZ
Processed data for EXC
Processed data for QCOM
Processed data for WBA
Processed data for NOW
Processed data for ADM
Processed data for WRB
Processed data for VMC
Processed data for SLB
Processed data for KVUE
Processed data for NI
Processed data for ROST
Processed data for WY
Processed data for SWKS
Processed data for CTVA
Processed data for NDAQ
Processed data for DVA
Processed data for EQIX
Processed data for AZO
Processed data for CMI
Processed data for DHR
Processed data for KR
Processed data for NTAP
Processed data for FOX
Processed data for ABBV
Processed data for FFIV
Processed data for RMD
Processed data for EG
Processed data for EA
Processed data for TMO
Processed data for MA
Processed data for UDR
Processed data for BF-B
```

```
Processed data for ALL
Processed data for CLX
Processed data for AME
Processed data for HWM
Processed data for A
Processed data for TGT
Processed data for WYNN
Processed data for BIIB
Processed data for OXY
Processed data for EIX
Processed data for GPN
Processed data for ANSS
Processed data for ADBE
Processed data for PANW
Processed data for HOLX
Processed data for BKR
Processed data for HLT
Processed data for WAT
Processed data for COF
Processed data for EPAM
Processed data for DPZ
Processed data for DAL
Processed data for MCHP
Processed data for LRCX
Processed data for WM
Processed data for FRT
Processed data for GE
Processed data for ACN
Processed data for DFS
Processed data for NKE
Processed data for CB
Processed data for INTC
Processed data for STZ
Processed data for CRM
Processed data for KO
Processed data for AMP
Processed data for T
Processed data for HBAN
Processed data for MKTX
Processed data for D
Processed data for RVTY
Processed data for ULTA
Processed data for L
Processed data for STLD
Processed data for LUV
Processed data for VLTO
Processed data for VICI
Processed data for TRV
```

```
Processed data for HIG
Processed data for BKNG
Processed data for CRWD
Processed data for IFF
Processed data for HUBB
Processed data for ZTS
Processed data for FDX
Processed data for TAP
Processed data for CMCSA
Processed data for TDY
Processed data for WST
Processed data for INTU
Processed data for MSI
Processed data for LEN
Processed data for CTAS
Processed data for LKQ
Processed data for COP
Processed data for AVB
Processed data for DECK
Processed data for AIG
Processed data for AOS
Processed data for AMTM
Processed data for PPG
Processed data for CAG
Processed data for DG
Processed data for BX
Processed data for CDNS
Processed data for USB
Processed data for MU
Processed data for TRMB
Processed data for AMZN
Processed data for EQR
Processed data for TSN
Processed data for HD
Processed data for RL
Processed data for GPC
Processed data for CEG
Processed data for CHRW
Processed data for PODD
Processed data for KEY
Processed data for HAS
Processed data for CHTR
Processed data for LULU
Processed data for DXCM
Processed data for TPL
Processed data for LHX
Processed data for C
Processed data for AWK
```

```
Processed data for PEG
Processed data for BR
Processed data for SBUX
Processed data for KEYS
Processed data for WBD
Processed data for DRI
Processed data for IQV
Processed data for VTRS
Processed data for ANET
Processed data for EMN
Processed data for GNRC
Processed data for MSFT
Processed data for VLO
Processed data for J
Processed data for DGX
Processed data for CTSH
Processed data for BALL
Processed data for WTW
Processed data for ADI
Processed data for GOOG
Processed data for NEE
Processed data for COR
Processed data for BRK-B
Processed data for DAY
Processed data for HAL
Processed data for O
Processed data for DIS
Processed data for FIS
Processed data for AMT
Processed data for YUM
Processed data for BEN
Processed data for CI
Processed data for XYL
Processed data for AES
Processed data for FITB
Processed data for VST
Processed data for PLTR
Processed data for MGM
Processed data for STE
Processed data for TYL
Processed data for ELV
Processed data for ZBRA
Processed data for ADSK
Processed data for PEP
Processed data for FDS
Processed data for AJG
Processed data for AMAT
Processed data for WFC
```

```
Processed data for AMGN
Processed data for TPR
Processed data for SBAC
Processed data for ES
Processed data for MTB
Processed data for DHI
Processed data for PG
Processed data for EW
Processed data for DOW
Processed data for CVS
Processed data for CRL
Processed data for ARE
Processed data for DD
Processed data for FSLR
Processed data for DOV
Processed data for CCL
Processed data for GD
Processed data for CAT
Processed data for MCO
Processed data for DTE
Processed data for CZR
Processed data for IP
Processed data for PHM
Processed data for BMY
Processed data for NSC
Processed data for WMT
Processed data for DE
Processed data for CAH
Processed data for NOC
Processed data for REGN
Processed data for MPC
Processed data for SWK
Processed data for JNPR
Processed data for TXN
Processed data for LH
Processed data for SHW
Processed data for CARR
Processed data for WDC
Processed data for PH
Processed data for SYF
Processed data for EVRG
Processed data for BLK
Processed data for DLR
Processed data for UBER
Processed data for ATO
Processed data for PNW
Processed data for TSLA
Processed data for OMC
```

```
Processed data for HRL
Processed data for BBY
Processed data for KMX
Processed data for CMG
Processed data for XOM
Processed data for TDG
Processed data for FMC
Processed data for JBL
Processed data for NDSN
Processed data for COST
Processed data for HCA
Processed data for ODFL
Processed data for RF
Processed data for IT
Processed data for FTNT
Processed data for SMCI
Processed data for ABT
Processed data for SPGI
Processed data for MO
Processed data for MRNA
Processed data for PTC
Processed data for EXPE
Processed data for IR
Processed data for ETR
Processed data for CSCO
Processed data for TROW
Processed data for FOXA
Processed data for NWSA
Processed data for NUE
Processed data for PGR
Processed data for LVS
Processed data for NXPI
Processed data for WAB
Processed data for DUK
Processed data for LMT
Processed data for AEE
Processed data for INVH
Processed data for V
Processed data for SW
Processed data for GIS
Processed data for ORCL
Processed data for GEV
Processed data for AIZ
Processed data for STX
Processed data for TT
Processed data for JBHT
Processed data for ED
Processed data for HSY
```

```
Processed data for GDDY
Processed data for TJX
Processed data for EL
Processed data for SOLV
Processed data for GEN
Processed data for PNC
Processed data for MTD
Processed data for PCG
Processed data for PLD
Processed data for REG
Processed data for UAL
Processed data for SPG
Processed data for NTRS
Processed data for SO
Processed data for CSX
Processed data for EQT
Processed data for IBM
Processed data for MRK
Processed data for BAC
Processed data for QRVO
Processed data for EBAY
Processed data for PAYX
Processed data for JKHY
Processed data for AXON
Processed data for ALGN
Processed data for DVN
Processed data for FCX
Processed data for VZ
Processed data for GRMN
Processed data for TSCO
Processed data for PFE
Processed data for HES
Processed data for CF
Processed data for URI
Processed data for GLW
Processed data for TMUS
Processed data for PSX
Processed data for CCI
Processed data for RTX
Processed data for EXPD
Processed data for WMB
Processed data for HII
Processed data for FTV
Processed data for CINF
Processed data for PPL
Processed data for SNA
Processed data for AEP
Processed data for GILD
```

```
Processed data for AFL
Processed data for ALB
Processed data for KIM
Processed data for MMC
Processed data for META
Processed data for EMR
Processed data for AXP
Processed data for BLDR
Processed data for ADP
Processed data for PRU
Processed data for CPB
Processed data for OTIS
Processed data for BRO
Processed data for GEHC
Processed data for MET
Processed data for BG
Processed data for BAX
Processed data for INCY
Processed data for POOL
Processed data for AVGO
Processed data for XEL
Processed data for CDW
Processed data for PFG
Processed data for UNH
Processed data for TTWO
Processed data for LNT
Processed data for ACGL
Processed data for HPQ
Processed data for HSIC
Processed data for ROK
Processed data for CVX
Processed data for PYPL
Processed data for TFC
Processed data for CL
Processed data for RSG
Processed data for CME
Processed data for KMI
Processed data for BDX
Processed data for ECL
Processed data for IPG
Processed data for TXT
Processed data for CNP
Processed data for LOW
Processed data for AKAM
Processed data for PKG
Processed data for EFX
Processed data for APTV
Processed data for PSA
```

```
Processed data for STT
Processed data for VRSK
Processed data for APH
Processed data for ESS
Processed data for KLAC
Processed data for KDP
Processed data for CNC
Processed data for LDOS
Processed data for PARA
Processed data for HON
Processed data for LYB
Processed data for MNST
Processed data for EXR
Processed data for AVY
Processed data for SNPS
Processed data for ICE
Processed data for KHC
Processed data for VRSN
Processed data for SCHW
Processed data for ALLE
Processed data for ERIE
Processed data for WELL
Processed data for ENPH
Processed data for EOG
Processed data for MOS
Processed data for CHD
Processed data for WEC
Processed data for UPS
Processed data for ZBH
Processed data for CFG
Processed data for CSGP
Processed data for OKE
Processed data for FAST
Processed data for VRTX
Processed data for PM
Processed data for MAA
Processed data for IDXX
Processed data for BA
Processed data for KMB
Processed data for LIN
Processed data for AMD
Processed data for PNR
Processed data for NEM
```

#Generate Vector Embeddings for RAG

[7]:
```
!pip install sentence-transformers
```

```
Requirement already satisfied: sentence-transformers in
/usr/local/lib/python3.10/dist-packages (3.2.1)
Requirement already satisfied: transformers<5.0.0,>=4.41.0 in
/usr/local/lib/python3.10/dist-packages (from sentence-transformers) (4.46.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages
(from sentence-transformers) (4.66.6)
Requirement already satisfied: torch>=1.11.0 in /usr/local/lib/python3.10/dist-
packages (from sentence-transformers) (2.5.1+cu121)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-
packages (from sentence-transformers) (1.5.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages
(from sentence-transformers) (1.13.1)
Requirement already satisfied: huggingface-hub>=0.20.0 in
/usr/local/lib/python3.10/dist-packages (from sentence-transformers) (0.26.3)
Requirement already satisfied: Pillow in /usr/local/lib/python3.10/dist-packages
(from sentence-transformers) (11.0.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-
packages (from huggingface-hub>=0.20.0->sentence-transformers) (3.16.1)
Requirement already satisfied: fsspec>=2023.5.0 in
/usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.20.0->sentence-
transformers) (2024.10.0)
Requirement already satisfied: packaging>=20.9 in
/usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.20.0->sentence-
transformers) (24.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-
packages (from huggingface-hub>=0.20.0->sentence-transformers) (6.0.2)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-
packages (from huggingface-hub>=0.20.0->sentence-transformers) (2.32.3)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.20.0->sentence-
transformers) (4.12.2)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-
packages (from torch>=1.11.0->sentence-transformers) (3.4.2)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages
(from torch>=1.11.0->sentence-transformers) (3.1.4)
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.10/dist-
packages (from torch>=1.11.0->sentence-transformers) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from
sympy==1.13.1->torch>=1.11.0->sentence-transformers) (1.3.0)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-
packages (from transformers<5.0.0,>=4.41.0->sentence-transformers) (1.26.4)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.10/dist-packages (from
transformers<5.0.0,>=4.41.0->sentence-transformers) (2024.9.11)
Requirement already satisfied: tokenizers<0.21,>=0.20 in
/usr/local/lib/python3.10/dist-packages (from
transformers<5.0.0,>=4.41.0->sentence-transformers) (0.20.3)
```

```
Requirement already satisfied: safetensors>=0.4.1 in
/usr/local/lib/python3.10/dist-packages (from
transformers<5.0.0,>=4.41.0->sentence-transformers) (0.4.5)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-
packages (from scikit-learn->sentence-transformers) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn->sentence-
transformers) (3.5.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.11.0->sentence-
transformers) (3.0.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface-
hub>=0.20.0->sentence-transformers) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests->huggingface-hub>=0.20.0->sentence-transformers) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface-
hub>=0.20.0->sentence-transformers) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface-
hub>=0.20.0->sentence-transformers) (2024.8.30)
```

[8]:
```python
from sentence_transformers import SentenceTransformer

# Initialize embedding model
embedding_model = SentenceTransformer('all-MiniLM-L6-v2')

# Create text descriptions for embeddings
def generate_embedding_descriptions(df):
    descriptions = []
    for _, row in df.iterrows():
        desc = (f"{row['Ticker']} had a mean close price of {row['Mean_Close']:.
 ↪2f}, "
                f"with a maximum close of {row['Max_Close']:.2f}, minimum close␣
 ↪of {row['Min_Close']:.2f}, "
                f"and a 3-month return of {row['Return_3m']:.2%}. "
                f"The latest close price was {row['Latest_Close']:.2f} on␣
 ↪{row['Latest_Date']} "
                f"with a trading volume of {row['Latest_Volume']}.")
        descriptions.append((row['Ticker'], desc))
    return descriptions

# Create embeddings
descriptions = generate_embedding_descriptions(processed_financial_data)
tickers, text_descriptions = zip(*descriptions)
embeddings = embedding_model.encode(text_descriptions)
```

```
# Save embeddings to a file
import numpy as np
np.save("financial_embeddings.npy", embeddings)
np.save("financial_tickers.npy", tickers)

print("Embeddings and tickers saved.")
```

```
/usr/local/lib/python3.10/dist-
packages/sentence_transformers/cross_encoder/CrossEncoder.py:13:
TqdmExperimentalWarning: Using `tqdm.autonotebook.tqdm` in notebook mode. Use
`tqdm.tqdm` instead to force console mode (e.g. in jupyter console)
  from tqdm.autonotebook import tqdm, trange
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_auth.py:94:
UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab
(https://huggingface.co/settings/tokens), set it as secret in your Google Colab
and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access
public models or datasets.
  warnings.warn(
modules.json:   0%|          | 0.00/349 [00:00<?, ?B/s]
config_sentence_transformers.json:   0%|          | 0.00/116 [00:00<?, ?B/s]
README.md:   0%|          | 0.00/10.7k [00:00<?, ?B/s]
sentence_bert_config.json:   0%|          | 0.00/53.0 [00:00<?, ?B/s]
config.json:   0%|          | 0.00/612 [00:00<?, ?B/s]
model.safetensors:   0%|          | 0.00/90.9M [00:00<?, ?B/s]
tokenizer_config.json:   0%|          | 0.00/350 [00:00<?, ?B/s]
vocab.txt:   0%|          | 0.00/232k [00:00<?, ?B/s]
tokenizer.json:   0%|          | 0.00/466k [00:00<?, ?B/s]
special_tokens_map.json:   0%|          | 0.00/112 [00:00<?, ?B/s]
1_Pooling/config.json:   0%|          | 0.00/190 [00:00<?, ?B/s]
Embeddings and tickers saved.
```

#Create vector database by storing vector embeddings using Pinecone

[9]:
```
!pip install -U pinecone-client
```

```
Collecting pinecone-client
  Downloading pinecone_client-5.0.1-py3-none-any.whl.metadata (19 kB)
```

```
Requirement already satisfied: certifi>=2019.11.17 in
/usr/local/lib/python3.10/dist-packages (from pinecone-client) (2024.8.30)
Collecting pinecone-plugin-inference<2.0.0,>=1.0.3 (from pinecone-client)
  Downloading pinecone_plugin_inference-1.1.0-py3-none-any.whl.metadata (2.2 kB)
Collecting pinecone-plugin-interface<0.0.8,>=0.0.7 (from pinecone-client)
  Downloading pinecone_plugin_interface-0.0.7-py3-none-any.whl.metadata (1.2 kB)
Requirement already satisfied: tqdm>=4.64.1 in /usr/local/lib/python3.10/dist-
packages (from pinecone-client) (4.66.6)
Requirement already satisfied: typing-extensions>=3.7.4 in
/usr/local/lib/python3.10/dist-packages (from pinecone-client) (4.12.2)
Requirement already satisfied: urllib3>=1.26.0 in
/usr/local/lib/python3.10/dist-packages (from pinecone-client) (2.2.3)
Downloading pinecone_client-5.0.1-py3-none-any.whl (244 kB)
                         244.8/244.8 kB
5.6 MB/s eta 0:00:00
Downloading pinecone_plugin_inference-1.1.0-py3-none-any.whl (85 kB)
                         85.4/85.4 kB
5.8 MB/s eta 0:00:00
Downloading pinecone_plugin_interface-0.0.7-py3-none-any.whl (6.2 kB)
Installing collected packages: pinecone-plugin-interface, pinecone-plugin-
inference, pinecone-client
Successfully installed pinecone-client-5.0.1 pinecone-plugin-inference-1.1.0
pinecone-plugin-interface-0.0.7
```

[10]:
```python
import os
from pinecone import Pinecone, ServerlessSpec
import numpy as np
import pandas as pd

# Load saved embeddings and tickers
embeddings = np.load("financial_embeddings.npy")
tickers = np.load("financial_tickers.npy", allow_pickle=True)

# Load the processed financial data CSV for metadata
processed_df = pd.read_csv("processed_financial_data.csv")
processed_data = {row['Ticker']: row.to_dict() for _, row in processed_df.
 ↪iterrows()}

# Initialize Pinecone
pc =␣
 ↪Pinecone(api_key="pcsk_61Bjni_Qpge2ZniRoyXPZwzExHqo6BkWXGjUGem4wMER7SRKsdGaafmpmXEUAGUcc4cH

# Create or connect to the index
index_name = "finance-index"

# Check if the index exists
if index_name not in [idx.name for idx in pc.list_indexes()]:
```

```
    pc.create_index(
        name=index_name,
        dimension=embeddings.shape[1],
        metric="cosine",
        spec=ServerlessSpec(
            cloud="aws",
            region="us-east-1"
        )
    )

# Connect to the index
index = pc.Index(index_name)

# Prepare and upsert vectors with metadata
vectors_to_upsert = []
for ticker, embedding in zip(tickers, embeddings):
    # Extract relevant metadata from the DataFrame
    metadata = processed_data.get(ticker, {})

    # Format for upsert
    vector = {
        "id": str(ticker),
        "values": embedding.tolist(),
        "metadata": {
            "Latest_Close": metadata.get("Latest_Close", "N/A"),
            "Return_3m": metadata.get("Return_3m", "N/A"),
            "Latest_Volume": metadata.get("Latest_Volume", "N/A"),
            "Latest_Date": metadata.get("Latest_Date", "N/A"),
        }
    }
    vectors_to_upsert.append(vector)

# Perform upsert
index.upsert(vectors=vectors_to_upsert)

print("Embeddings with metadata uploaded to Pinecone.")
```

```
Embeddings with metadata uploaded to Pinecone.
```

#Query Vector Database (RAG)

[51]:
```
from sentence_transformers import SentenceTransformer

# Initialize the embedding model
embedding_model = SentenceTransformer('all-MiniLM-L6-v2')

# Query Pinecone Function
def query_pinecone(user_query, top_k=5):
```

```
    # Generate embedding for the query
    query_embedding = embedding_model.encode(user_query).tolist()

    # Search in Pinecone
    search_results = index.query(
        vector=query_embedding,
        top_k=top_k,
        include_metadata=True  # Include data fields
    )

    return search_results
```

[52]:
```
# Test the query system
test_query = "What is the latest performance of AAPL stock?"
search_results=query_pinecone(test_query)
print(search_results)
```

```
{'matches': [{'id': 'AAPL',
              'metadata': {'Latest_Close': 247.75030517578125,
                           'Latest_Date': '2024-12-10',
                           'Latest_Volume': 18279599.0,
                           'Return_3m': 0.1126843667520409},
              'score': 0.680406451,
              'values': []},
             {'id': 'ABT',
              'metadata': {'Latest_Close': 115.48999786376952,
                           'Latest_Date': '2024-12-10',
                           'Latest_Volume': 1566470.0,
                           'Return_3m': -0.0095197309166169},
              'score': 0.501015723,
              'values': []},
             {'id': 'AOS',
              'metadata': {'Latest_Close': 73.08499908447266,
                           'Latest_Date': '2024-12-10',
                           'Latest_Volume': 334991.0,
                           'Return_3m': -0.0685062460198743},
              'score': 0.468043029,
              'values': []},
             {'id': 'AON',
              'metadata': {'Latest_Close': 359.7449951171875,
                           'Latest_Date': '2024-12-10',
                           'Latest_Volume': 553191.0,
                           'Return_3m': 0.043343990293852},
              'score': 0.466181695,
              'values': []},
             {'id': 'APD',
              'metadata': {'Latest_Close': 314.79998779296875,
```

```
                            'Latest_Date': '2024-12-10',
                            'Latest_Volume': 328845.0,
                            'Return_3m': 0.1335565787300462},
               'score': 0.45967862,
               'values': []}],
 'namespace': '',
 'usage': {'read_units': 6}}
```

[13]:
```python
# Correct the metadata extraction logic
def extract_financial_data_from_pinecone(search_results):
    # Ensure matches exist
    if search_results and 'matches' in search_results and␣
 ↪len(search_results['matches']) > 0:
        best_match = search_results['matches'][0]  # First result
        ticker = best_match['id']

        # Check if metadata exists
        if 'metadata' in best_match:
            metadata = best_match['metadata']
            # Format the response using metadata fields
            response = (
                f"{ticker} stock's latest performance:\n"
                f"- Latest Close Price: ${metadata.get('Latest_Close', 'N/A'):.
 ↪2f}\n"
                f"- 3-Month Return: {metadata.get('Return_3m', 'N/A'):.2%}\n"
                f"- Latest Trading Volume: {metadata.get('Latest_Volume', 'N/
 ↪A')}\n"
                f"- Date: {metadata.get('Latest_Date', 'N/A')}\n"
            )
            return response
        else:
            return f"Ticker {ticker} found, but no financial data is available."

    return "Sorry, no relevant financial data could be retrieved at this time."
```

[14]:
```python
response = extract_financial_data_from_pinecone(search_results)
print(response)
```

```
AAPL stock's latest performance:
- Latest Close Price: $243.60
- 3-Month Return: 10.67%
- Latest Trading Volume: 2830922.0
- Date: 2024-12-09
```

Successfully retrieving desired records from vector DB

#Large Language Model 1 - Gemini 1.5 Flash

####**Implementing Virtual Assistant using Gemini with RAG pipeline**

- Introducing a standardized prompt for the LLM to generate desired response.

[1]:
```
!pip install google-generativeai
```

```
Requirement already satisfied: google-generativeai in
/usr/local/lib/python3.10/dist-packages (0.8.3)
Requirement already satisfied: google-ai-generativelanguage==0.6.10 in
/usr/local/lib/python3.10/dist-packages (from google-generativeai) (0.6.10)
Requirement already satisfied: google-api-core in
/usr/local/lib/python3.10/dist-packages (from google-generativeai) (2.19.2)
Requirement already satisfied: google-api-python-client in
/usr/local/lib/python3.10/dist-packages (from google-generativeai) (2.151.0)
Requirement already satisfied: google-auth>=2.15.0 in
/usr/local/lib/python3.10/dist-packages (from google-generativeai) (2.27.0)
Requirement already satisfied: protobuf in /usr/local/lib/python3.10/dist-
packages (from google-generativeai) (4.25.5)
Requirement already satisfied: pydantic in /usr/local/lib/python3.10/dist-
packages (from google-generativeai) (2.10.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages
(from google-generativeai) (4.66.6)
Requirement already satisfied: typing-extensions in
/usr/local/lib/python3.10/dist-packages (from google-generativeai) (4.12.2)
Requirement already satisfied: proto-plus<2.0.0dev,>=1.22.3 in
/usr/local/lib/python3.10/dist-packages (from google-ai-
generativelanguage==0.6.10->google-generativeai) (1.25.0)
Requirement already satisfied: googleapis-common-protos<2.0.dev0,>=1.56.2 in
/usr/local/lib/python3.10/dist-packages (from google-api-core->google-
generativeai) (1.66.0)
Requirement already satisfied: requests<3.0.0.dev0,>=2.18.0 in
/usr/local/lib/python3.10/dist-packages (from google-api-core->google-
generativeai) (2.32.3)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from google-auth>=2.15.0->google-
generativeai) (5.5.0)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/usr/local/lib/python3.10/dist-packages (from google-auth>=2.15.0->google-
generativeai) (0.4.1)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.10/dist-
packages (from google-auth>=2.15.0->google-generativeai) (4.9)
Requirement already satisfied: httplib2<1.dev0,>=0.19.0 in
/usr/local/lib/python3.10/dist-packages (from google-api-python-client->google-
generativeai) (0.22.0)
Requirement already satisfied: google-auth-httplib2<1.0.0,>=0.2.0 in
/usr/local/lib/python3.10/dist-packages (from google-api-python-client->google-
generativeai) (0.2.0)
Requirement already satisfied: uritemplate<5,>=3.0.1 in
```

```
/usr/local/lib/python3.10/dist-packages (from google-api-python-client->google-
generativeai) (4.1.1)
Requirement already satisfied: annotated-types>=0.6.0 in
/usr/local/lib/python3.10/dist-packages (from pydantic->google-generativeai)
(0.7.0)
Requirement already satisfied: pydantic-core==2.27.1 in
/usr/local/lib/python3.10/dist-packages (from pydantic->google-generativeai)
(2.27.1)
Requirement already satisfied: grpcio<2.0dev,>=1.33.2 in
/usr/local/lib/python3.10/dist-packages (from google-api-core[grpc]!=2.0.*,!=2.1
.*,!=2.10.*,!=2.2.*,!=2.3.*,!=2.4.*,!=2.5.*,!=2.6.*,!=2.7.*,!=2.8.*,!=2.9.*,<3.0
.0dev,>=1.34.1->google-ai-generativelanguage==0.6.10->google-generativeai)
(1.68.1)
Requirement already satisfied: grpcio-status<2.0.dev0,>=1.33.2 in
/usr/local/lib/python3.10/dist-packages (from google-api-core[grpc]!=2.0.*,!=2.1
.*,!=2.10.*,!=2.2.*,!=2.3.*,!=2.4.*,!=2.5.*,!=2.6.*,!=2.7.*,!=2.8.*,!=2.9.*,<3.0
.0dev,>=1.34.1->google-ai-generativelanguage==0.6.10->google-generativeai)
(1.62.3)
Requirement already satisfied:
pyparsing!=3.0.0,!=3.0.1,!=3.0.2,!=3.0.3,<4,>=2.4.2 in
/usr/local/lib/python3.10/dist-packages (from httplib2<1.dev0,>=0.19.0->google-
api-python-client->google-generativeai) (3.2.0)
Requirement already satisfied: pyasn1<0.7.0,>=0.4.6 in
/usr/local/lib/python3.10/dist-packages (from pyasn1-modules>=0.2.1->google-
auth>=2.15.0->google-generativeai) (0.6.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from
requests<3.0.0.dev0,>=2.18.0->google-api-core->google-generativeai) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests<3.0.0.dev0,>=2.18.0->google-api-core->google-
generativeai) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from
requests<3.0.0.dev0,>=2.18.0->google-api-core->google-generativeai) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from
requests<3.0.0.dev0,>=2.18.0->google-api-core->google-generativeai) (2024.8.30)
```

[16]:
```python
import google.generativeai as genai

# Replace 'YOUR_API_KEY' with your actual API key
genai.configure(api_key='AIzaSyAovQlg-UAnqIp_NdNZ_6ntX6yIDm13HYs')
```

[20]:
```python
def generate_response_with_gemini(user_query, top_k=5):
    # Retrieve financial data
    search_results = query_pinecone(user_query, top_k)
    extracted_data = extract_financial_data_from_pinecone(search_results)
```

```
    # Handle missing data case
    if "No relevant financial data found" in extracted_data:
        return "Sorry, I couldn't find relevant financial data for your query."

    # Create a detailed prompt
    prompt = (
        f"Based on the following financial data retrieved from our database:\n"
        f"{extracted_data}\n\n"
        f"User Question :{user_query}\n"
        f"Please provide a detailed, professional answer summarizing the␣
↪stock's latest "
        f"performance, including any relevant analysis such as trends, recent␣
↪activity, or "
        f"market conditions."
    )

    # Generate response using Gemini
    model = genai.GenerativeModel("gemini-1.5-flash")
    response = model.generate_content(prompt)

    return response.text
```

###Gemini Response for sample question

[21]:
```
user_query = "What is the latest performance of AAPL stock?"
response = generate_response_with_gemini(user_query)
print("Response:", response)
```

```
Response: As of December 9th, 2024, Apple (AAPL) stock closed at $243.60.  Over
the past three months, the stock has exhibited a strong performance, yielding a
10.67% return.  The latest trading volume of 2,830,922 shares suggests moderate
trading activity on this particular day.

While a single day's closing price and volume provide a snapshot, the 3-month
return of 10.67% indicates a positive trend in AAPL's performance during that
period.  To provide a more comprehensive analysis, additional data points would
be needed.  This includes:

* **Longer-term performance:** Examining returns over 6-month, 1-year, and
5-year periods would reveal broader trends and help contextualize the recent
3-month gain.  A comparison to the overall market performance (e.g., S&P 500)
during these periods would also be valuable in assessing AAPL's relative
strength.
* **Volatility:** Measuring the standard deviation of returns over different
time periods would quantify the risk associated with AAPL's price fluctuations.
High volatility suggests greater price swings and risk.
```

```
* **Market conditions:** Understanding the broader macroeconomic environment
(interest rates, inflation, economic growth) and sector-specific factors
(technology industry trends) is crucial to interpret the performance.  Positive
market conditions could contribute to the positive 3-month return, while
negative conditions might suggest AAPL is outperforming the market.
* **Analyst ratings and price targets:** Consulting analyst reports and their
price targets for AAPL would offer further insights into future expectations for
the stock.
* **News and events:** Reviewing recent news and events affecting AAPL (e.g.,
product launches, earnings announcements, regulatory changes) would help explain
the performance.

In conclusion, the provided data shows positive recent performance for AAPL, but
a complete picture requires a more in-depth analysis encompassing the factors
listed above.  The 10.67% 3-month return is a strong indication of positive
momentum, but further investigation is necessary to assess its sustainability
and underlying drivers.
```

[15]:
```
from huggingface_hub import notebook_login

notebook_login()
```

```
VBox(children=(HTML(value='<center> <img\nsrc=https://huggingface.co/front/
↪assets/huggingface_logo-noborder.sv…
```

[16]:
```
!pip install transformers accelerate
```

```
Requirement already satisfied: transformers in /usr/local/lib/python3.10/dist-
packages (4.46.3)
Requirement already satisfied: accelerate in /usr/local/lib/python3.10/dist-
packages (1.1.1)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-
packages (from transformers) (3.16.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.23.2 in
/usr/local/lib/python3.10/dist-packages (from transformers) (0.26.3)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-
packages (from transformers) (1.26.4)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from transformers) (24.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-
packages (from transformers) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.10/dist-packages (from transformers) (2024.9.11)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-
packages (from transformers) (2.32.3)
Requirement already satisfied: tokenizers<0.21,>=0.20 in
/usr/local/lib/python3.10/dist-packages (from transformers) (0.20.3)
```

```
Requirement already satisfied: safetensors>=0.4.1 in
/usr/local/lib/python3.10/dist-packages (from transformers) (0.4.5)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-
packages (from transformers) (4.66.6)
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages
(from accelerate) (5.9.5)
Requirement already satisfied: torch>=1.10.0 in /usr/local/lib/python3.10/dist-
packages (from accelerate) (2.5.1+cu121)
Requirement already satisfied: fsspec>=2023.5.0 in
/usr/local/lib/python3.10/dist-packages (from huggingface-
hub<1.0,>=0.23.2->transformers) (2024.10.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.10/dist-packages (from huggingface-
hub<1.0,>=0.23.2->transformers) (4.12.2)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-
packages (from torch>=1.10.0->accelerate) (3.4.2)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages
(from torch>=1.10.0->accelerate) (3.1.4)
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.10/dist-
packages (from torch>=1.10.0->accelerate) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from
sympy==1.13.1->torch>=1.10.0->accelerate) (1.3.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests->transformers) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests->transformers)
(2024.8.30)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.10.0->accelerate)
(3.0.2)
```

#**Large Language Model 2 : Flan-t5-large (google)**

####**Implementing Virtual Assistant using Flan-t5-large with RAG pipeline**

- Introducing a standardized prompt for the LLM to generate desired response.

[17]:
```
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM
import torch

model_name = "google/flan-t5-large"

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
```

```
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSeq2SeqLM.from_pretrained(model_name, device_map="auto").
 ↪to(device)
```

```
tokenizer_config.json:   0%|          | 0.00/2.54k [00:00<?, ?B/s]
spiece.model:   0%|          | 0.00/792k [00:00<?, ?B/s]
tokenizer.json:   0%|          | 0.00/2.42M [00:00<?, ?B/s]
special_tokens_map.json:   0%|          | 0.00/2.20k [00:00<?, ?B/s]
config.json:   0%|          | 0.00/662 [00:00<?, ?B/s]
model.safetensors:   0%|          | 0.00/3.13G [00:00<?, ?B/s]
generation_config.json:   0%|          | 0.00/147 [00:00<?, ?B/s]
```

[56]:
```
def generate_response_with_flan_t5(user_query, top_k=5):
    # Query Pinecone and extract data
    search_results = query_pinecone(user_query, top_k)
    extracted_data = extract_financial_data_from_pinecone(search_results)

    # Handle missing data case
    if "No relevant financial data found" in extracted_data:
        return "Sorry, I couldn't find relevant financial data for your query."

    # Create a detailed prompt
    prompt = (
        f"Based on the following financial data retrieved from our database:\n"
        f"{extracted_data}\n\n"
        f"{user_query}\n"
        f"Please provide a detailed, professional answer summarizing the␣
 ↪stock's latest "
        f"performance, including any relevant analysis such as trends, recent␣
 ↪activity, or "
        f"market conditions."
    )

    # Tokenize and generate response
    inputs = tokenizer(prompt, return_tensors="pt", truncation=True).to(device)
    # Generate response
    output = model.generate(
        **inputs,
        max_length=1024,              # Allow long responses
        num_beams=5,                  # Use beam search for better answers
        length_penalty=1.0,           # Avoid very short answers
        early_stopping=True           # Stop when done
    )
```

```
    # Decode the Response
    response = tokenizer.decode(output[0], skip_special_tokens=True)
    return response
```

###Flan-t5-large model's response for sample questions

[19]:
```
query = "What is the latest performance of AAPL stock?"
response = generate_response_with_flan_t5(query)
print("Response from the Model:")
print(response)
```

```
Response from the Model:
The latest close price of AAPL stock was $247.75 and the 3-Month return was
11.27%. The stock has a current trading volume of 18279599.0.
```

[63]:
```
query = "How has TSLA performed over the last 3 months?"
response = generate_response_with_flan_t5(query)
print("Response from the Model:")
print(response)
```

```
Response from the Model:
TSLA stock has returned 78.29% over the last 3 months.
```

#**Gradio UI**

- Used Flan-t5-large model

[20]:
```
!pip install gradio
```

```
Collecting gradio
  Downloading gradio-5.8.0-py3-none-any.whl.metadata (16 kB)
Collecting aiofiles<24.0,>=22.0 (from gradio)
  Downloading aiofiles-23.2.1-py3-none-any.whl.metadata (9.7 kB)
Requirement already satisfied: anyio<5.0,>=3.0 in
/usr/local/lib/python3.10/dist-packages (from gradio) (3.7.1)
Collecting fastapi<1.0,>=0.115.2 (from gradio)
  Downloading fastapi-0.115.6-py3-none-any.whl.metadata (27 kB)
Collecting ffmpy (from gradio)
  Downloading ffmpy-0.4.0-py3-none-any.whl.metadata (2.9 kB)
Collecting gradio-client==1.5.1 (from gradio)
  Downloading gradio_client-1.5.1-py3-none-any.whl.metadata (7.1 kB)
Requirement already satisfied: httpx>=0.24.1 in /usr/local/lib/python3.10/dist-
packages (from gradio) (0.28.0)
Requirement already satisfied: huggingface-hub>=0.25.1 in
/usr/local/lib/python3.10/dist-packages (from gradio) (0.26.3)
Requirement already satisfied: jinja2<4.0 in /usr/local/lib/python3.10/dist-
packages (from gradio) (3.1.4)
Collecting markupsafe~=2.0 (from gradio)
  Downloading MarkupSafe-2.1.5-cp310-cp310-
```

```
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (3.0 kB)
Requirement already satisfied: numpy<3.0,>=1.0 in
/usr/local/lib/python3.10/dist-packages (from gradio) (1.26.4)
Requirement already satisfied: orjson~=3.0 in /usr/local/lib/python3.10/dist-
packages (from gradio) (3.10.12)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-
packages (from gradio) (24.2)
Requirement already satisfied: pandas<3.0,>=1.0 in
/usr/local/lib/python3.10/dist-packages (from gradio) (2.2.2)
Requirement already satisfied: pillow<12.0,>=8.0 in
/usr/local/lib/python3.10/dist-packages (from gradio) (11.0.0)
Requirement already satisfied: pydantic>=2.0 in /usr/local/lib/python3.10/dist-
packages (from gradio) (2.10.3)
Collecting pydub (from gradio)
  Downloading pydub-0.25.1-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting python-multipart>=0.0.18 (from gradio)
  Downloading python_multipart-0.0.19-py3-none-any.whl.metadata (1.8 kB)
Requirement already satisfied: pyyaml<7.0,>=5.0 in
/usr/local/lib/python3.10/dist-packages (from gradio) (6.0.2)
Collecting ruff>=0.2.2 (from gradio)
  Downloading ruff-0.8.2-py3-none-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (25 kB)
Collecting safehttpx<0.2.0,>=0.1.6 (from gradio)
  Downloading safehttpx-0.1.6-py3-none-any.whl.metadata (4.2 kB)
Collecting semantic-version~=2.0 (from gradio)
  Downloading semantic_version-2.10.0-py2.py3-none-any.whl.metadata (9.7 kB)
Collecting starlette<1.0,>=0.40.0 (from gradio)
  Downloading starlette-0.41.3-py3-none-any.whl.metadata (6.0 kB)
Collecting tomlkit<0.14.0,>=0.12.0 (from gradio)
  Downloading tomlkit-0.13.2-py3-none-any.whl.metadata (2.7 kB)
Requirement already satisfied: typer<1.0,>=0.12 in
/usr/local/lib/python3.10/dist-packages (from gradio) (0.15.0)
Requirement already satisfied: typing-extensions~=4.0 in
/usr/local/lib/python3.10/dist-packages (from gradio) (4.12.2)
Collecting uvicorn>=0.14.0 (from gradio)
  Downloading uvicorn-0.32.1-py3-none-any.whl.metadata (6.6 kB)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages
(from gradio-client==1.5.1->gradio) (2024.10.0)
Collecting websockets<15.0,>=10.0 (from gradio-client==1.5.1->gradio)
  Downloading websockets-14.1-cp310-cp310-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl.metadata (6.7 kB)
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.10/dist-
packages (from anyio<5.0,>=3.0->gradio) (3.10)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.10/dist-
packages (from anyio<5.0,>=3.0->gradio) (1.3.1)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-
packages (from anyio<5.0,>=3.0->gradio) (1.2.2)
```

```
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-
packages (from httpx>=0.24.1->gradio) (2024.8.30)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.10/dist-
packages (from httpx>=0.24.1->gradio) (1.0.7)
Requirement already satisfied: h11<0.15,>=0.13 in
/usr/local/lib/python3.10/dist-packages (from
httpcore==1.*->httpx>=0.24.1->gradio) (0.14.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-
packages (from huggingface-hub>=0.25.1->gradio) (3.16.1)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-
packages (from huggingface-hub>=0.25.1->gradio) (2.32.3)
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.10/dist-
packages (from huggingface-hub>=0.25.1->gradio) (4.66.6)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas<3.0,>=1.0->gradio) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
packages (from pandas<3.0,>=1.0->gradio) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-
packages (from pandas<3.0,>=1.0->gradio) (2024.2)
Requirement already satisfied: annotated-types>=0.6.0 in
/usr/local/lib/python3.10/dist-packages (from pydantic>=2.0->gradio) (0.7.0)
Requirement already satisfied: pydantic-core==2.27.1 in
/usr/local/lib/python3.10/dist-packages (from pydantic>=2.0->gradio) (2.27.1)
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.10/dist-
packages (from typer<1.0,>=0.12->gradio) (8.1.7)
Requirement already satisfied: shellingham>=1.3.0 in
/usr/local/lib/python3.10/dist-packages (from typer<1.0,>=0.12->gradio) (1.5.4)
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.10/dist-
packages (from typer<1.0,>=0.12->gradio) (13.9.4)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-
packages (from python-dateutil>=2.8.2->pandas<3.0,>=1.0->gradio) (1.16.0)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/usr/local/lib/python3.10/dist-packages (from
rich>=10.11.0->typer<1.0,>=0.12->gradio) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/usr/local/lib/python3.10/dist-packages (from
rich>=10.11.0->typer<1.0,>=0.12->gradio) (2.18.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface-
hub>=0.25.1->gradio) (3.4.0)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface-
hub>=0.25.1->gradio) (2.2.3)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-
packages (from markdown-it-py>=2.2.0->rich>=10.11.0->typer<1.0,>=0.12->gradio)
(0.1.2)
Downloading gradio-5.8.0-py3-none-any.whl (57.2 MB)
                           57.2/57.2 MB
```

```
11.6 MB/s eta 0:00:00
Downloading gradio_client-1.5.1-py3-none-any.whl (320 kB)
                         320.2/320.2 kB
21.0 MB/s eta 0:00:00
Downloading aiofiles-23.2.1-py3-none-any.whl (15 kB)
Downloading fastapi-0.115.6-py3-none-any.whl (94 kB)
                         94.8/94.8 kB
6.6 MB/s eta 0:00:00
Downloading
MarkupSafe-2.1.5-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (25
kB)
Downloading python_multipart-0.0.19-py3-none-any.whl (24 kB)
Downloading ruff-0.8.2-py3-none-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(11.2 MB)
                         11.2/11.2 MB
94.9 MB/s eta 0:00:00
Downloading safehttpx-0.1.6-py3-none-any.whl (8.7 kB)
Downloading semantic_version-2.10.0-py2.py3-none-any.whl (15 kB)
Downloading starlette-0.41.3-py3-none-any.whl (73 kB)
                         73.2/73.2 kB
5.5 MB/s eta 0:00:00
Downloading tomlkit-0.13.2-py3-none-any.whl (37 kB)
Downloading uvicorn-0.32.1-py3-none-any.whl (63 kB)
                         63.8/63.8 kB
5.1 MB/s eta 0:00:00
Downloading ffmpy-0.4.0-py3-none-any.whl (5.8 kB)
Downloading pydub-0.25.1-py2.py3-none-any.whl (32 kB)
Downloading websockets-14.1-cp310-cp310-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl (168 kB)
                         168.2/168.2 kB
11.9 MB/s eta 0:00:00
Installing collected packages: pydub, websockets, uvicorn, tomlkit,
semantic-version, ruff, python-multipart, markupsafe, ffmpy, aiofiles,
starlette, safehttpx, gradio-client, fastapi, gradio
  Attempting uninstall: markupsafe
    Found existing installation: MarkupSafe 3.0.2
    Uninstalling MarkupSafe-3.0.2:
      Successfully uninstalled MarkupSafe-3.0.2
Successfully installed aiofiles-23.2.1 fastapi-0.115.6 ffmpy-0.4.0 gradio-5.8.0
gradio-client-1.5.1 markupsafe-2.1.5 pydub-0.25.1 python-multipart-0.0.19
ruff-0.8.2 safehttpx-0.1.6 semantic-version-2.10.0 starlette-0.41.3
tomlkit-0.13.2 uvicorn-0.32.1 websockets-14.1
```

[21]:
```
import gradio as gr

# Example Function Connecting Backend Logic
```

```
def get_stock_performance(query):
    # Query Pinecone and Generate Answer (Replace with Your Existing Function)
    response = generate_response_with_flan_t5(query)
    return response

# Create a Beautiful Gradio UI
with gr.Blocks() as demo:
    # Header Section
    gr.Markdown(""
        """
        #  **Finance Chatbot**
        Ask questions about **stock performance**, **market trends**, and␣
 ↪**financial data analysis**.
        """
    )

    # Create Input and Output Sections
    user_input = gr.Textbox(
        label="Ask Your Financial Question",
        placeholder="e.g., What is the latest performance of AAPL stock?",
        lines=2,
        interactive=True  # Make input field dynamic
    )

    submit_btn = gr.Button("Ask")

    # Output Section
    output_text = gr.Textbox(
        label="Model's Response",
        placeholder="The answer will appear here...",
        lines=6
    )

    # Link the Submit Button to Backend Function
    submit_btn.click(fn=get_stock_performance, inputs=user_input,␣
 ↪outputs=output_text)

# Launch the Gradio UI
demo.launch()
```

```
Running Gradio in a Colab notebook requires sharing enabled. Automatically
setting `share=True` (you can turn this off by setting `share=False` in
`launch()` explicitly).

Colab notebook detected. To show errors in colab notebook, set debug=True in
launch()
* Running on public URL: https://46e01108b38b26e00f.gradio.live
```

```
This share link expires in 72 hours. For free permanent hosting and GPU
upgrades, run `gradio deploy` from the terminal in the working directory to
deploy to Hugging Face Spaces (https://huggingface.co/spaces)
```

```
<IPython.core.display.HTML object>
```

[21]:

#**Evaluation**

- Used flan-t5-large
- Challenge Faced: Gemini quota getting exceeded

####Set/Define 30 Evaluation Questions

[79]:
```
import pandas as pd
import numpy as np
from sklearn.metrics import accuracy_score, precision_score, recall_score

# Define 30 Evaluation Questions
questions = [
    "What is the latest performance of AAPL stock?",
    "How has TSLA performed over the last 3 months?",
    "What is the latest performance of MSFT stock?",
    "Can you summarize the recent activity of GOOGL stock?",
    "How is AMZN stock performing currently?",
    "What is the latest closing price of AAPL stock?",
    "What was the highest closing price of TSLA in the last 3 months?",
    "What was the lowest closing price of MSFT recently?",
    "How does GOOGL's latest close compare to 3 months ago?",
    "What is the current price of AMZN stock?",
    "What is the latest trading volume for AAPL stock?",
    "How many shares of TSLA were traded most recently?",
    "What was the average trading volume of MSFT over the last 3 months?",
    "How does GOOGL's trading volume compare to AMZN's?",
    "What was the highest trading volume for TSLA in the past 3 months?",
    "What is the 3-month return of AAPL stock?",
    "How much has TSLA's stock value changed in the last 3 months?",
    "What is the percentage return of MSFT over the past quarter?",
    "How does the 3-month return of GOOGL compare to AAPL's?",
    "What is AMZN's stock return over the last 3 months?",
    "Which stock performed better in the last 3 months: AAPL or TSLA?",
    "Compare the 3-month returns of MSFT and GOOGL.",
    "Which company had the highest trading volume recently: AMZN or TSLA?",
    "How does AAPL's latest performance compare to its average over the past 3␣
 ↪months?",
    "Is GOOGL's 3-month return higher than AMZN's?",
    "What was the closing price of AAPL stock 3 months ago?",
```

```
    "What is the historical trend for TSLA stock over the last 3 months?",
    "Can you provide the last 3 closing prices for MSFT stock?",
    "How has GOOGL's price changed week over week in the last month?",
    "What were the 3-month trading volumes for AMZN stock?",
]
```

####Generate ground truth for all the questions

[80]:
```
import re

def get_ground_truth(question):
    # Identify the ticker symbol from the question
    if "AAPL" in question:
        ticker = "AAPL"
    elif "TSLA" in question:
        ticker = "TSLA"
    elif "MSFT" in question:
        ticker = "MSFT"
    elif "GOOGL" in question:
        ticker = "GOOGL"
    elif "AMZN" in question:
        ticker = "AMZN"
    else:
        return "Ground truth not available"

    # Get the data for the identified ticker
    row = financial_data[financial_data["Ticker"] == ticker]
    if row.empty:
        return "Ground truth not available"

    # Extract relevant data based on the type of question
    # Adaptive Ground Truth Logic
    if re.search(r"\bperform", question.lower()):
        return (
            f"The latest close price of {ticker} is ${row['Latest_Close'].
 ↪values[0]:.2f}, "
            f"with a 3-month return of {row['Return_3m'].values[0]:.2%}, "
            f"and a trading volume of {row['Latest_Volume'].values[0]}."
        )

    elif re.search(r"\b(price|closing price)", question.lower()):
        return f"The latest closing price of {ticker} is ${row['Latest_Close'].
 ↪values[0]:.2f}."

    elif re.search(r"\btrading volume", question.lower()):
        return f"The latest trading volume of {ticker} is {row['Latest_Volume'].
 ↪values[0]}."
```

```
    elif re.search(r"\breturn|3-month|3 month", question.lower()):
        return f"The 3-month return of {ticker} is {row['Return_3m'].values[0]:.
↪2%}."

    elif re.search(r"\b(compare|better)", question.lower()):
        return (
            f"{ticker} stock's latest metrics: Closing Price:␣
↪${row['Latest_Close'].values[0]:.2f}, "
            f"3-Month Return: {row['Return_3m'].values[0]:.2%}, "
            f"Trading Volume: {row['Latest_Volume'].values[0]}."
        )
    else:
      # Default Case for Other Queries
      return (
            f"The latest close price of {ticker} is ${row['Latest_Close'].
↪values[0]:.2f}, "
            f"with a 3-month return of {row['Return_3m'].values[0]:.2%}, "
            f"and a trading volume of {row['Latest_Volume'].values[0]}."
        )
```

####Generate model response for all the 30 questions

[81]:
```
def generate_response(query):
    response = generate_response_with_flan_t5(query)
    return response
```

####Compare ground truth and model response using cosine similarity

[82]:
```
from sentence_transformers import SentenceTransformer
from sklearn.metrics.pairwise import cosine_similarity

# Load Pre-trained Sentence Transformer
embedding_model = SentenceTransformer('all-MiniLM-L6-v2')

# Function to Calculate Cosine Similarity
def cosine_similarity_score(response, ground_truth):
    # Generate embeddings
    response_emb = embedding_model.encode(response, convert_to_tensor=True)
    ground_truth_emb = embedding_model.encode(ground_truth,␣
↪convert_to_tensor=True)

    # Calculate cosine similarity
    similarity = cosine_similarity(response_emb.unsqueeze(0), ground_truth_emb.
↪unsqueeze(0))[0][0]
    return similarity
```

####Evaluate with cosine similarity

[83]:
```
# Updated Evaluation Function with Cosine Similarity
def evaluate_question_with_cosine(question):
    ground_truth = get_ground_truth(question)
    model_response = generate_response_with_flan_t5(question)

    # Calculate Cosine Similarity
    cosine_similarity_val = cosine_similarity_score(model_response,␣
 ↪ground_truth)

    # Consider correct if cosine similarity > 0.7
    is_correct = cosine_similarity_val > 0.7

    return {
        "question": question,
        "ground_truth": ground_truth,
        "model_response": model_response,
        "cosine_similarity": cosine_similarity_val,
        "is_correct": is_correct
    }
```

[84]:
```
# Run Evaluation for All Questions
results_with_cosine = [evaluate_question_with_cosine(q) for q in questions]

# Convert Results to DataFrame
results_with_cosine_df = pd.DataFrame(results_with_cosine)
```

####Evaluation metrics and visualization

[85]:
```
# Metrics Calculation
accuracy_cosine = results_with_cosine_df["is_correct"].mean()
average_cosine_similarity = results_with_cosine_df["cosine_similarity"].mean()

# Print Metrics
print("Evaluation Results with Cosine Similarity:")
print(results_with_cosine_df[["question", "ground_truth", "model_response",␣
 ↪"cosine_similarity", "is_correct"]])
print(f"\nOverall Accuracy (Cosine): {accuracy_cosine * 100:.2f}%")
print(f"Average Cosine Similarity: {average_cosine_similarity:.2f}")
```

```
Evaluation Results with Cosine Similarity:
                                              question  \
0       What is the latest performance of AAPL stock?
1      How has TSLA performed over the last 3 months?
2       What is the latest performance of MSFT stock?
3   Can you summarize the recent activity of GOOGL…
4             How is AMZN stock performing currently?
5     What is the latest closing price of AAPL stock?
6   What was the highest closing price of TSLA in …
```

```
7   What was the lowest closing price of MSFT rece…
8   How does GOOGL's latest close compare to 3 mon…
9            What is the current price of AMZN stock?
10  What is the latest trading volume for AAPL stock?
11  How many shares of TSLA were traded most recen…
12  What was the average trading volume of MSFT ov…
13  How does GOOGL's trading volume compare to AMZ…
14  What was the highest trading volume for TSLA i…
15          What is the 3-month return of AAPL stock?
16  How much has TSLA's stock value changed in the…
17  What is the percentage return of MSFT over the…
18  How does the 3-month return of GOOGL compare t…
19  What is AMZN's stock return over the last 3 mo…
20  Which stock performed better in the last 3 mon…
21     Compare the 3-month returns of MSFT and GOOGL.
22  Which company had the highest trading volume r…
23  How does AAPL's latest performance compare to …
24      Is GOOGL's 3-month return higher than AMZN's?
25  What was the closing price of AAPL stock 3 mon…
26  What is the historical trend for TSLA stock ov…
27  Can you provide the last 3 closing prices for …
28  How has GOOGL's price changed week over week i…
29  What were the 3-month trading volumes for AMZN…

                                          ground_truth  \
0   The latest close price of AAPL is $247.75, wit…
1   The latest close price of TSLA is $406.73, wit…
2   The latest close price of MSFT is $444.23, wit…
3   The latest close price of GOOGL is $184.57, wi…
4   The latest close price of AMZN is $225.63, wit…
5        The latest closing price of AAPL is $247.75.
6        The latest closing price of TSLA is $406.73.
7        The latest closing price of MSFT is $444.23.
8              The 3-month return of GOOGL is 22.10%.
9        The latest closing price of AMZN is $225.63.
10   The latest trading volume of AAPL is 18279599.0.
11  The latest close price of TSLA is $406.73, wit…
12    The latest trading volume of MSFT is 8066770.0.
13  The latest trading volume of GOOGL is 37286206.0.
14   The latest trading volume of TSLA is 70894686.0.
15             The 3-month return of AAPL is 11.27%.
16             The 3-month return of TSLA is 78.29%.
17              The 3-month return of MSFT is 5.01%.
18             The 3-month return of AAPL is 11.27%.
19             The 3-month return of AMZN is 22.28%.
20  The latest close price of AAPL is $247.75, wit…
21              The 3-month return of MSFT is 5.01%.
22   The latest trading volume of TSLA is 70894686.0.
```

```
23  The latest close price of AAPL is $247.75, wit…
24             The 3-month return of GOOGL is 22.10%.
25       The latest closing price of AAPL is $247.75.
26              The 3-month return of TSLA is 78.29%.
27       The latest closing price of MSFT is $444.23.
28      The latest closing price of GOOGL is $184.57.
29   The latest trading volume of AMZN is 17373460.0.

                                       model_response  cosine_similarity  \
0   The latest close price of AAPL stock was $247…           0.959086
1   TSLA stock has returned 78.29% over the last 3…           0.670150
2   The latest close price of MSFT stock was $444…           0.952956
3   GOOGL has a 3-month return of 22.10% and a cur…           0.888633
4   The stock has a 3-month return of 22.28% and a…           0.536378
5   The latest closing price of AAPL stock was $24…           0.943775
6   $406.73 was the highest closing price of TSLA …           0.913727
7   The latest closing price of MSFT stock was $44…           0.938103
8   GOOGL's latest close price was $184.57 and its…           0.794186
9         The current price of AMZN stock is $225.63.             0.860560
10  The latest trading volume for AAPL stock is 18…           0.964851
11  The stock has a 3-month return of 78.29%. The …           0.429140
12  The average volume of MSFT over the last 3 mon…           0.718691
13  GOOGL has a 3-month return of 22.10% and a tra…           0.809683
14  70894686.0 was the highest volume traded for T…           0.848304
15        The 3-month return of AAPL stock is 11.27%.             0.940478
16  TSLA's stock has changed 78.29% in the last 3 …           0.803259
17          MSFT stock has a 3-month return of 5.01%.             0.910694
18  GOOGL's 3-month return is 11.27% compared to A…           0.763654
19  AMZN stock has returned 22.28% over the last 3…           0.880022
20               AAPL has a 3-month return of 11.27%.             0.684981
21  GOOGL has a 3-month return of 22.10% and a tra…           0.510786
22               AMZN has a 3-month return of 22.28%.             0.212964
23  The stock's average return over the past 3 mon…           0.428252
24                                              Yes             0.126927
25  The latest closing price of AAPL stock was $24…           0.911988
26          The stock has a 3-month return of 78.29%.             0.636317
27  The latest closing price of MSFT stock was $44…           0.899995
28  GOOGL has a 3-Month return of 22.10% and a tra…           0.631068
29  The latest trading volume for AMZN stock was 1…           0.957112

    is_correct
0         True
1        False
2         True
3         True
4        False
5         True
6         True
```

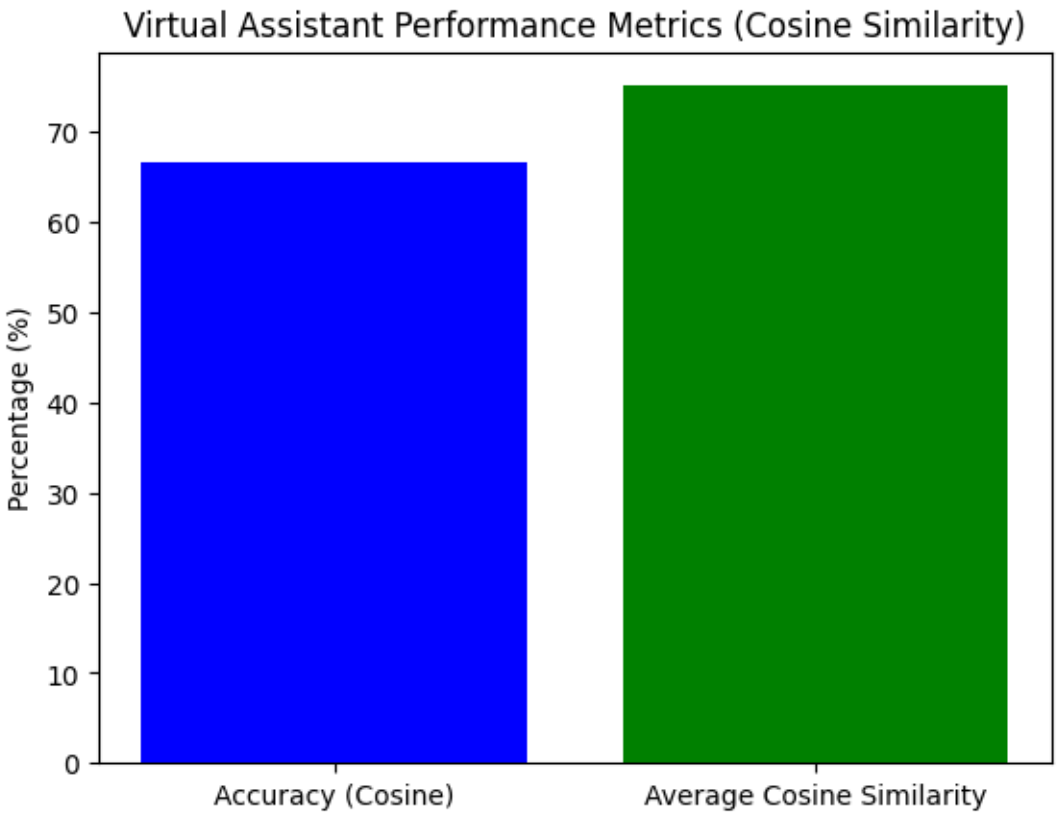
```
7         True
8         True
9         True
10        True
11       False
12        True
13        True
14        True
15        True
16        True
17        True
18        True
19        True
20       False
21       False
22       False
23       False
24       False
25        True
26       False
27        True
28       False
29        True

Overall Accuracy (Cosine): 66.67%
Average Cosine Similarity: 0.75
```

[86]:
```
print(f"\nOverall Accuracy (Cosine): {accuracy_cosine * 100:.2f}%")
print(f"Average Cosine Similarity: {average_cosine_similarity * 100:.2f}%")
```

```
Overall Accuracy (Cosine): 66.67%
Average Cosine Similarity: 75.09%
```

[87]:
```
# Visualization
plt.bar(["Accuracy (Cosine)", "Average Cosine Similarity"], [accuracy_cosine *␣
 ↪100, average_cosine_similarity * 100], color=["blue", "green"])
plt.title("Virtual Assistant Performance Metrics (Cosine Similarity)")
plt.ylabel("Percentage (%)")
plt.show()
```



#Improving Performance of Virtual Assistant

###Introducing Adaptive Prompting, revising VA response and applied filtering

####Introducing Adaptive prompting

[98]:
```
import re

def create_adaptive_prompt(user_query, extracted_data):
    """
    Creates an adaptive prompt with sentence-based instructions for␣
 ↪completeness.
    """

    # Sentence-Enforced Adaptive Instructions
    if re.search(r"\bperform", user_query.lower()):
        instruction = (
            f"Please provide a complete, professional sentence summarizing the␣
 ↪stock's latest performance, "
            f"including its closing price, 3-month return, trading volume, and␣
 ↪any relevant market analysis."
        )
```

```
    elif re.search(r"\b(price|closing price)", user_query.lower()):
        instruction = (
            f"Please provide a complete sentence stating the stock's latest␣
↪closing price, "
            f"mentioning the ticker symbol and the specific closing value."
        )

    elif re.search(r"\btrading volume", user_query.lower()):
        instruction = (
            f"Please provide a sentence describing the stock's latest trading␣
↪volume, "
            f"including the trading date, stock symbol, and the exact number of␣
↪shares traded."
        )

    elif re.search(r"\breturn", user_query.lower()):
        instruction = (
            f"Please provide a sentence stating the stock's 3-month return␣
↪percentage, "
            f"including its ticker symbol and precise return value."
        )

    elif re.search(r"\b(compare|better)", user_query.lower()):
        instruction = (
            f"Please compare the stocks' recent performances in a clear␣
↪sentence based on their closing prices, "
            f"3-month returns, and trading volumes, mentioning the␣
↪better-performing stock."
        )

    else:
        instruction = (
            f"Please provide a professional summary of the stock's key␣
↪financial metrics, including closing price, "
            f"returns, trading volume, and any relevant market context in a␣
↪complete sentence."
        )

    # Final Context-Aware Prompt
    prompt = (
        f"Based on the following financial data retrieved from our system:\n\n"
        f"{extracted_data}\n\n"
        f"User Question: {user_query}\n\n"
        f"{instruction}\n"
    )
```

```
    return prompt
```

####Generating model response with improved prompting

[99]:
```
def generate_response_with_flan_t5_adaptive_prompting(user_query, top_k=5):
    # Query Pinecone and extract data
    search_results = query_pinecone(user_query, top_k)
    extracted_data = extract_financial_data_from_pinecone(search_results)

    # Handle missing data case
    if "No relevant financial data found" in extracted_data:
        return "Sorry, I couldn't find relevant financial data for your query."

    prompt= create_adaptive_prompt(user_query, extracted_data)

    # Tokenize and generate response
    inputs = tokenizer(prompt, return_tensors="pt", truncation=True).to(device)
    # Generate response
    output = model.generate(
        **inputs,
        max_length=1024,              # Allow long responses
        num_beams=5,                  # Use beam search for better answers
        length_penalty=1.0,           # Avoid very short answers
        early_stopping=True           # Stop when done
    )

    # Decode the Response
    response = tokenizer.decode(output[0], skip_special_tokens=True)
    return response
```

[92]:
```
query = "What is the latest performance of AAPL stock?"
response = generate_response_with_flan_t5_adaptive_prompting(query)
print("Response from the Model:")
print(response)
```

```
Response from the Model:
The latest closing price of AAPL stock was $247.75 and its 3-month return was
11.27%. The stock has a recent trading volume of 18279599.0.
```

[91]:
```
query = "How has TSLA performed over the last 3 months?"
response = generate_response_with_flan_t5_adaptive_prompting(query)
print("Response from the Model:")
print(response)
```

```
Response from the Model:
The stock has a 3-month return of 78.29% and a recent trading volume of
70894686.0.
```

####Revising Virtual Assistant Response

[101]:
```
def reformat_response(question, response):
    """
    Reformat model responses if only numerical values are returned.
    """
    if re.match(r"^\$?\d+(\.\d+)?%?$", response.strip()):
        # Response is a number; reformat based on query type
        if re.search(r"\b(price|closing price)", question.lower()):
            return f"The latest closing price of the stock is {response.
 ↪strip()}."

        elif re.search(r"\btrading volume", question.lower()):
            return f"The latest trading volume of the stock is {response.
 ↪strip()}."

        elif re.search(r"\breturn|3 month|3-month", question.lower()):
            return f"The 3-month return of the stock is {response.strip()}."

    # Return Original Response if Already Properly Formatted
    return response.strip()
```

####Apply Filtering - Grammar Checker

[103]:
```
# Install and Initialize Language Tool
!pip install language-tool-python
import language_tool_python

# Initialize Grammar Checker
tool = language_tool_python.LanguageTool('en-US')

# Filter Responses Using Grammar Check
def filter_response(response):
    corrected_response = tool.correct(response)
    return corrected_response
```

```
Collecting language-tool-python
  Downloading language_tool_python-2.8.1-py3-none-any.whl.metadata (12 kB)
Requirement already satisfied: pip in /usr/local/lib/python3.10/dist-packages
(from language-tool-python) (24.1.2)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-
packages (from language-tool-python) (2.32.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages
(from language-tool-python) (4.66.6)
Requirement already satisfied: wheel in /usr/local/lib/python3.10/dist-packages
(from language-tool-python) (0.45.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests->language-tool-python)
```

```
(3.4.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests->language-tool-python) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests->language-tool-python)
(2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests->language-tool-python)
(2024.8.30)
Downloading language_tool_python-2.8.1-py3-none-any.whl (35 kB)
Installing collected packages: language-tool-python
Successfully installed language-tool-python-2.8.1

Downloading LanguageTool 6.4: 100%|      | 246M/246M [00:06<00:00, 40.2MB/s]
INFO:language_tool_python.download_lt:Unzipping /tmp/tmp39mnarwi.zip to
/root/.cache/language_tool_python.
INFO:language_tool_python.download_lt:Downloaded
https://www.languagetool.org/download/LanguageTool-6.4.zip to
/root/.cache/language_tool_python.
```

####Evaluate with cosine similarity

####Compare ground truth and improved model response

[104]:
```python
# Updated Evaluation Function with Cosine Similarity
def evaluate_question_with_cosine(question):
    ground_truth = get_ground_truth(question)
    model_response = generate_response_with_flan_t5_adaptive_prompting(question)
    # Reformat Incomplete Responses
    # Revising VA response text prior to showing it to the user
    corrected_response = reformat_response(question, model_response)

    # Post-process with Grammar Correction
    corrected_response = filter_response(corrected_response)

    # Calculate Cosine Similarity
    cosine_similarity_val = cosine_similarity_score(corrected_response,␣
 ↪ground_truth)

    # Consider correct if cosine similarity > 0.7
    is_correct = cosine_similarity_val > 0.7

    return {
        "question": question,
        "ground_truth": ground_truth,
        "model_response": corrected_response,
        "cosine_similarity": cosine_similarity_val,
        "is_correct": is_correct
    }
```

[105]:
```
# Run Evaluation for All Questions
results_with_cosine = [evaluate_question_with_cosine(q) for q in questions]

# Convert Results to DataFrame
results_with_cosine_df = pd.DataFrame(results_with_cosine)
```

####Evaluation Metrics and Visualization

[106]:
```
# Metrics Calculation
accuracy_cosine = results_with_cosine_df["is_correct"].mean()
average_cosine_similarity = results_with_cosine_df["cosine_similarity"].mean()

# Print Metrics
print("Evaluation Results with Cosine Similarity:")
print(results_with_cosine_df[["question", "ground_truth", "model_response",␣
 ↪"cosine_similarity", "is_correct"]])
print(f"\nOverall Accuracy (Cosine): {accuracy_cosine * 100:.2f}%")
print(f"Average Cosine Similarity: {average_cosine_similarity:.2f}")
```

```
Evaluation Results with Cosine Similarity:
                                              question  \
0       What is the latest performance of AAPL stock?
1      How has TSLA performed over the last 3 months?
2       What is the latest performance of MSFT stock?
3   Can you summarize the recent activity of GOOGL…
4             How is AMZN stock performing currently?
5     What is the latest closing price of AAPL stock?
6   What was the highest closing price of TSLA in …
7   What was the lowest closing price of MSFT rece…
8   How does GOOGL's latest close compare to 3 mon…
9           What is the current price of AMZN stock?
10  What is the latest trading volume for AAPL stock?
11  How many shares of TSLA were traded most recen…
12  What was the average trading volume of MSFT ov…
13  How does GOOGL's trading volume compare to AMZ…
14  What was the highest trading volume for TSLA i…
15         What is the 3-month return of AAPL stock?
16  How much has TSLA's stock value changed in the…
17  What is the percentage return of MSFT over the…
18  How does the 3-month return of GOOGL compare t…
19  What is AMZN's stock return over the last 3 mo…
20  Which stock performed better in the last 3 mon…
21     Compare the 3-month returns of MSFT and GOOGL.
22  Which company had the highest trading volume r…
23  How does AAPL's latest performance compare to …
24      Is GOOGL's 3-month return higher than AMZN's?
25  What was the closing price of AAPL stock 3 mon…
26  What is the historical trend for TSLA stock ov…
```

```
27  Can you provide the last 3 closing prices for …
28  How has GOOGL's price changed week over week i…
29  What were the 3-month trading volumes for AMZN…

                                          ground_truth  \
0   The latest close price of AAPL is $247.75, wit…
1   The latest close price of TSLA is $406.73, wit…
2   The latest close price of MSFT is $444.23, wit…
3   The latest close price of GOOGL is $184.57, wi…
4   The latest close price of AMZN is $225.63, wit…
5        The latest closing price of AAPL is $247.75.
6        The latest closing price of TSLA is $406.73.
7        The latest closing price of MSFT is $444.23.
8              The 3-month return of GOOGL is 22.10%.
9        The latest closing price of AMZN is $225.63.
10  The latest trading volume of AAPL is 18279599.0.
11  The latest close price of TSLA is $406.73, wit…
12   The latest trading volume of MSFT is 8066770.0.
13  The latest trading volume of GOOGL is 37286206.0.
14  The latest trading volume of TSLA is 70894686.0.
15             The 3-month return of AAPL is 11.27%.
16             The 3-month return of TSLA is 78.29%.
17              The 3-month return of MSFT is 5.01%.
18             The 3-month return of AAPL is 11.27%.
19             The 3-month return of AMZN is 22.28%.
20  The latest close price of AAPL is $247.75, wit…
21              The 3-month return of MSFT is 5.01%.
22  The latest trading volume of TSLA is 70894686.0.
23  The latest close price of AAPL is $247.75, wit…
24             The 3-month return of GOOGL is 22.10%.
25       The latest closing price of AAPL is $247.75.
26              The 3-month return of TSLA is 78.29%.
27       The latest closing price of MSFT is $444.23.
28      The latest closing price of GOOGL is $184.57.
29  The latest trading volume of AMZN is 17373460.0.

                                      model_response  cosine_similarity  \
0   AAPL stock's latest performance, including its…           0.885402
1   The stock TSLA has a 3-month return of 78.29%,…           0.899864
2   MSFT stock's latest performance, including its…           0.868436
3   GOOGLE stock closed at $184.57 on 2024-12-10, …           0.672570
4   The latest closing price of AMZN stock was $22…           0.948299
5   AAPL stock's latest performance: - Latest Clos…           0.831292
6   TSLA stock's latest performance: Latest Close …           0.828089
7   MSFT stock has a recent closing price of $444.23.           0.920747
8   GOOGLE stock's latest close price was $184.57 …           0.585514
9   The latest closing price of AMZN stock was $22…           0.946401
10  The latest trading volume for AAPL stock was 1…           0.946556
```

11	TSLA stock's latest performance: - Latest Clos…	0.878180
12	MSFT stock has a 3-month return of 5.01%. The …	0.674922
13	GOOGLE has a trading volume of 37286206.0 shar…	0.676569
14	TSLA stock's latest performance: - Latest Clos…	0.689828
15	AAPL stock has a 3-month return of 11.27%, its…	0.846759
16	TSLA stock's latest performance: - Latest Clos…	0.700637
17	MSFT stock's latest performance: - Latest Clos…	0.648773
18	GOOGLE has a 3-month return of 11.27%.	0.605600
19	AMZN stock has a 3-month return of 22.28%, its…	0.862514
20	AAPL stock has a 3-month return of 11.27% and …	0.887384
21	GOOGLE has a 3-month return of 22.10%, its tic…	0.565793
22	AMZN has the highest volume with 17373460.0 sh…	0.481239
23	AAPL stock's latest performance, including its…	0.788612
24	GOOGLE stock has a 3-month return of 22.10%.	0.703213
25	AAPL stock's latest performance: - Latest Clos…	0.832447
26	TSLA stock has a 3-month return of 78.29%.	0.893421
27	MSFT stock's latest closing price, mentioning …	0.890102
28	GOOGLE stock's latest closing price, mentionin…	0.761335
29	AMZN stock's latest performance: - Latest Clos…	0.745395

	is_correct
0	True
1	True
2	True
3	False
4	True
5	True
6	True
7	True
8	False
9	True
10	True
11	True
12	False
13	False
14	False
15	True
16	True
17	False
18	False
19	True
20	True
21	False
22	False
23	True
24	True
25	True
26	True

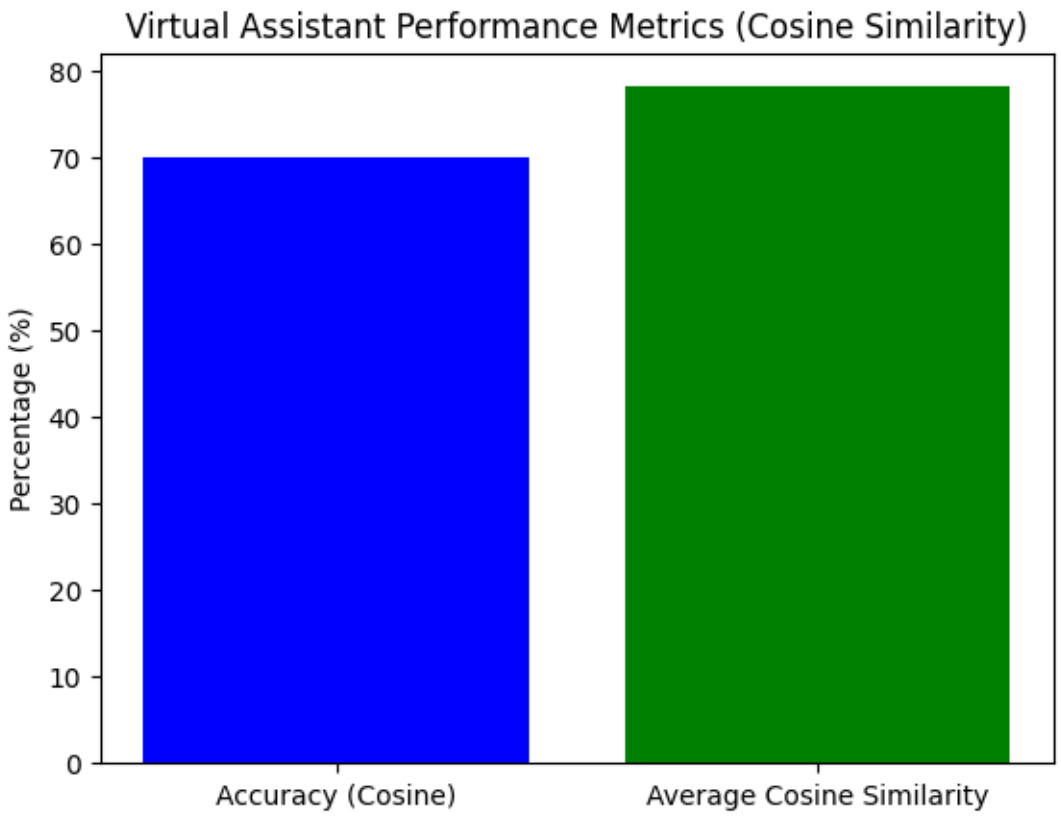
```
27        True
28        True
29        True

Overall Accuracy (Cosine): 70.00%
Average Cosine Similarity: 0.78
```

[107]:
```
print(f"\nOverall Accuracy (Cosine): {accuracy_cosine * 100:.2f}%")
print(f"Average Cosine Similarity: {average_cosine_similarity * 100:.2f}%")
```

```
Overall Accuracy (Cosine): 70.00%
Average Cosine Similarity: 78.22%
```

[108]:
```
# Visualization
plt.bar(["Accuracy (Cosine)", "Average Cosine Similarity"], [accuracy_cosine *␣
↪100, average_cosine_similarity * 100], color=["blue", "green"])
plt.title("Virtual Assistant Performance Metrics (Cosine Similarity)")
plt.ylabel("Percentage (%)")
plt.show()
```



Virtual Assistant's Performance Significantly improved after applying certain techniques (introduc-

ing adaptive prompting, revising VA response and apply filtering).

#**Challenges Faced So Far**

- **Creating embeddings as the data is numerical** - Overcame by converting each row of numerical data to a meaningful string representation to understand the financial data and then generating embeddings.
- **Choosing lightweight models**, as most of the models are either running for long time or not giving response since the query/prompt is quite large - Overcame by trying out more than 10 models and fixing the best ones.
- **Flan-t5-large was generating multiple or redundant responses** - Overcame by tweaking model parameters and applying early stopping.
- **Writing ground truth manually for evaluation**, as it is time consuming and repetitive - Overcame by generating adaptive ground truth based on the question.
- **Improving performance of VA.** Tried various methods individually and tested the performance and was still not able to improve - Finally was able to improve the performance by combining 3 different approaches together after numerous attempts.