import Dataset

```
!pip install yfinance
```

```
Requirement already satisfied: yfinance in /usr/local/lib/python3.10/dist-packages (0.2.40)
Requirement already satisfied: pandas>=1.3.0 in /usr/local/lib/python3.10/dist-packages (from yfinance) (2.0.3)
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) (2.31.0)
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) (4.9.4)
Requirement already satisfied: platformdirs>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from yfinance) (4.2.2)
Requirement already satisfied: pytz>=2022.5 in /usr/local/lib/python3.10/dist-packages (from yfinance) (2023.4)
Requirement already satisfied: frozendict>=2.3.4 in /usr/local/lib/python3.10/dist-packages (from yfinance) (2.4.4)
Requirement already satisfied: peewee>=3.16.2 in /usr/local/lib/python3.10/dist-packages (from yfinance) (3.17.5)
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: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (from beautifulsoup4>=4.11.1->yfinance) (2.5)
Requirement already satisfied: charset-normalizer>=2.0.1,<4 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance) (2024.7.4)
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>=0.4 in /usr/local/lib/python3.10/dist-packages (from html5lib>=1.1->yfinance) (0.5.1)
Note: you may need to restart the kernel to use updated packages.
```

```
import yfinance as yf
import pandas as pd
```

```
tickers = ['GAPL', 'TEXT', 'GEOS', 'MEDP', 'PFGC', 'TLX', 'UFI', 'ARCO', 'GNWW', 'CPF']

# Fetch data for the specified list of tickers
data = yf.download(tickers, start='2010-01-01')
print(data.head())

# Display the last few rows of the data
print(data.tail())
```

```
[*********************100%***********************]  10 of 10 completed
Price      Adj Close                                                   \
Ticker          ARCO  CPF  GAPL  GEOS  GNWW  MEDP  PFGC  TEXT  TLX  UFI
Date
2010-01-04  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00
2010-01-05  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00
2010-01-06  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00
2010-01-07  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00
2010-01-08  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00

Price      Close                                                       \
Ticker          ARCO  CPF  GAPL  GEOS  GNWW  MEDP  PFGC  TEXT  TLX  UFI
Date
2010-01-04  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00
2010-01-05  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00
2010-01-06  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00
2010-01-07  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00
2010-01-08  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00

Price      High                                                        \
Ticker          ARCO  CPF  GAPL  GEOS  GNWW  MEDP  PFGC  TEXT  TLX  UFI
Date
2010-01-04  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00
2010-01-05  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00
2010-01-06  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00
2010-01-07  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00
2010-01-08  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00

Price      Low                                                         \
Ticker          ARCO  CPF  GAPL  GEOS  GNWW  MEDP  PFGC  TEXT  TLX  UFI
Date
2010-01-04  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00
2010-01-05  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00
2010-01-06  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00
2010-01-07  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00
2010-01-08  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00

Price      Open                                                        \
Ticker          ARCO  CPF  GAPL  GEOS  GNWW  MEDP  PFGC  TEXT  TLX  UFI
Date
2010-01-04  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00
2010-01-05  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00
2010-01-06  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00
2010-01-07  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00
2010-01-08  19.800000  NaN  5.00  8.00  5.00  0.00  0.00  0.00  0.00  0.00

Price      Volume                                                      \
Ticker          ARCO  CPF  GAPL  GEOS  GNWW  MEDP  PFGC  TEXT  TLX  UFI
Date
2010-01-04     0.00  NaN  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
2010-01-05     0.00  NaN  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
2010-01-06     0.00  NaN  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
2010-01-07     0.00  NaN  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
2010-01-08     0.00  NaN  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00

[5 rows x 50 columns]
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 3774 entries, 2010-01-04 to 2025-12-29
Data columns (total 50 columns):
 #   Adj Close, ARCO   3774 non-null   float64
 #   Adj Close, CPF    1000 non-null   float64
 #   Adj Close, GAPL   3774 non-null   float64
 #   Adj Close, GEOS   3774 non-null   float64
 #   Adj Close, GNWW   3774 non-null   float64
 #   Adj Close, MEDP   3774 non-null   float64
 #   Adj Close, PFGC   3774 non-null   float64
 #   Adj Close, TEXT   3774 non-null   float64
 #   Adj Close, TLX    3774 non-null   float64
 #   Adj Close, UFI    3774 non-null   float64
 #   Close, ARCO       3774 non-null   float64
 #   Close, CPF        1000 non-null   float64
 #   Close, GAPL       3774 non-null   float64
 #   Close, GEOS       3774 non-null   float64
 #   Close, GNWW       3774 non-null   float64
 #   Close, MEDP       3774 non-null   float64
 #   Close, PFGC       3774 non-null   float64
 #   Close, TEXT       3774 non-null   float64
 #   Close, TLX        3774 non-null   float64
 #   Close, UFI        3774 non-null   float64
 #   High, ARCO        3774 non-null   float64
 #   High, CPF         1000 non-null   float64
 #   High, GAPL        3774 non-null   float64
 #   High, GEOS        3774 non-null   float64
 #   High, GNWW        3774 non-null   float64
 #   High, MEDP        3774 non-null   float64
 #   High, PFGC        3774 non-null   float64
 #   High, TEXT        3774 non-null   float64
 #   High, TLX         3774 non-null   float64
 #   High, UFI         3774 non-null   float64
 #   Low, ARCO         3774 non-null   float64
 #   Low, CPF          1000 non-null   float64
 #   Low, GAPL         3774 non-null   float64
 #   Low, GEOS         3774 non-null   float64
 #   Low, GNWW         3774 non-null   float64
 #   Low, MEDP         3774 non-null   float64
 #   Low, PFGC         3774 non-null   float64
 #   Low, TEXT         3774 non-null   float64
 #   Low, TLX          3774 non-null   float64
 #   Low, UFI          3774 non-null   float64
 #   Open, ARCO        3774 non-null   float64
 #   Open, CPF         1000 non-null   float64
 #   Open, GAPL        3774 non-null   float64
 #   Open, GEOS        3774 non-null   float64
 #   Open, GNWW        3774 non-null   float64
 #   Open, MEDP        3774 non-null   float64
 #   Open, PFGC        3774 non-null   float64
 #   Open, TEXT        3774 non-null   float64
 #   Open, TLX         3774 non-null   float64
 #   Open, UFI         3774 non-null   float64
 #   Volume, ARCO      3774 non-null   int64
 #   Volume, CPF       1000 non-null   int64
 #   Volume, GAPL      3774 non-null   int64
 #   Volume, GEOS      3774 non-null   int64
 #   Volume, GNWW      3774 non-null   int64
 #   Volume, MEDP      3774 non-null   int64
 #   Volume, PFGC      3774 non-null   int64
 #   Volume, TEXT      3774 non-null   int64
 #   Volume, TLX       3774 non-null   int64
 #   Volume, UFI       3774 non-null   int64
dtypes: float64(45), int64(5)
memory usage: 153.9 MB
```

cleaning the data

```
data.dropna()
```

```
[100%%********************100%%********************]  10 of 10 completed
```

```
# Check for missing values
print(data.isnull().sum())
```

```
Price  Adj Close  ARCO    0
                  CPF     0
                  GAPL    0
                  GEOS    0
                  GNWW    0
                  MEDP    0
                  PFGC    0
                  TEXT    0
                  TLX     0
                  UFI     0
Close            ARCO    0
                  CPF     0
                  GAPL    0
                  GEOS    0
                  GNWW    0
                  MEDP    0
                  PFGC    0
                  TEXT    0
                  TLX     0
                  UFI     0
High             ARCO    0
                  CPF     0
                  GAPL    0
                  GEOS    0
                  GNWW    0
                  MEDP    0
                  PFGC    0
                  TEXT    0
                  TLX     0
                  UFI     0
Low              ARCO    0
                  CPF     0
                  GAPL    0
                  GEOS    0
                  GNWW    0
                  MEDP    0
                  PFGC    0
                  TEXT    0
                  TLX     0
                  UFI     0
Open             ARCO    0
                  CPF     0
                  GAPL    0
                  GEOS    0
                  GNWW    0
                  MEDP    0
                  PFGC    0
                  TEXT    0
                  TLX     0
                  UFI     0
Volume           ARCO    0
                  CPF     0
                  GAPL    0
                  GEOS    0
                  GNWW    0
                  MEDP    0
                  PFGC    0
                  TEXT    0
                  TLX     0
                  UFI     0
dtype: int64
```

```
# Drop columns with any NaN values
data = data.dropna(axis=1, how='any')
```

```
# Check missing values
print(data.isnull().sum())
```

```
Price  Adj Close  ARCO    0
                  GAPL    0
                  GEOS    0
                  GNWW    0
                  MEDP    0
                  PFGC    0
                  TEXT    0
                  TLX     0
                  UFI     0
Close            ARCO    0
                  GAPL    0
                  GEOS    0
                  GNWW    0
                  MEDP    0
                  PFGC    0
                  TEXT    0
                  TLX     0
                  UFI     0
High             ARCO    0
                  GAPL    0
                  GEOS    0
                  GNWW    0
                  MEDP    0
                  PFGC    0
                  TEXT    0
                  TLX     0
                  UFI     0
Low              ARCO    0
                  GAPL    0
                  GEOS    0
                  GNWW    0
                  MEDP    0
                  PFGC    0
                  TEXT    0
                  TLX     0
                  UFI     0
Open             ARCO    0
                  GAPL    0
                  GEOS    0
                  GNWW    0
                  MEDP    0
                  PFGC    0
                  TEXT    0
                  TLX     0
                  UFI     0
Volume           ARCO    0
                  GAPL    0
                  GEOS    0
                  GNWW    0
                  MEDP    0
                  PFGC    0
                  TEXT    0
                  TLX     0
                  UFI     0
dtype: int64
```

```
# Check shape after dropping NaN values
print(data.shape)
```

```
(3774, 40)
```

```
# Define dependent variable
data['Target'] = (data['Close', 'GAPL'].shift(-1) > data['Close', 'GAPL'])
```

```
# Create multiple features for stock
data['Close', 'GAPL'].pct_change(5).dropna()
```

```
Price  Adj Close  ARCO    0
                  GAPL    0
                  GEOS    0
                  GNWW    0
                  MEDP    0
                  PFGC    0
                  TEXT    0
                  TLX     0
                  UFI     0
Close            ARCO    0
                  GAPL    0
                  GEOS    0
                  GNWW    0
                  MEDP    0
                  PFGC    0
                  TEXT    0
                  TLX     0
                  UFI     0
High             ARCO    0
                  GAPL    0
                  GEOS    0
                  GNWW    0
                  MEDP    0
                  PFGC    0
                  TEXT    0
                  TLX     0
                  UFI     0
Low              ARCO    0
                  GAPL    0
                  GEOS    0
                  GNWW    0
                  MEDP    0
                  PFGC    0
                  TEXT    0
                  TLX     0
                  UFI     0
Open             ARCO    0
                  GAPL    0
                  GEOS    0
                  GNWW    0
                  MEDP    0
                  PFGC    0
                  TEXT    0
                  TLX     0
                  UFI     0
Volume           ARCO    0
                  GAPL    0
                  GEOS    0
                  GNWW    0
                  MEDP    0
                  PFGC    0
                  TEXT    0
                  TLX     0
                  UFI     0
dtype: int64
```

Exploratory Data Analysis (EDA)

```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
tickers = ['GAPL', 'TEXT', 'GEOS', 'MEDP', 'PFGC', 'TLX', 'UFI', 'ARCO', 'GNWW', 'CPF']
```

Historical Adjusted Closing Prices

```
# Plot the data
sns.set(style='darkgrid')

# Plot historical closing prices
plt.figure(figure=(12, 6))
for ticker in tickers:
    plt.plot(data['Adj Close', ticker], label=ticker)

plt.title('Historical Adjusted Closing Prices')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()
```

Historical Trading Volumes

```
plt.figure(figure=(12, 6))
for ticker in tickers:
    plt.plot(data['Volume', ticker], label=ticker)

plt.title('Historical Trading Volumes')
plt.xlabel('Date')
plt.ylabel('Volume')
plt.legend()
plt.show()
```

Correlation Matrix of Daily Returns

```
# Calculate daily returns
daily_returns = data.pct_change()

# Calculate the correlation matrix
correlation_matrix = daily_returns.corr()

# Plot the correlation matrix
plt.figure(figure=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidth=0.5)
plt.title('Correlation Matrix of Daily Returns')
plt.show()
```

Moving Averages

```
# Calculate moving 30-day average
for ticker in tickers:
    data['MA30_' + ticker] = data['Adj Close', ticker].rolling(window=30).mean()
    data['MA90_' + ticker] = data['Adj Close', ticker].rolling(window=90).mean()

# Plot moving averages for one stock
plt.figure(figure=(12, 6))
plt.plot(data['Adj Close', 'GAPL'], label='GAPL')
plt.plot(data['MA30_GAPL'], label='MA30_GAPL', linestyle='--')
plt.plot(data['MA90_GAPL'], label='MA90_GAPL', linestyle='-.')
plt.title('GAPL Adjusted Close and Moving Averages')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()
```

Daily Volatility of Stocks

```
# Calculate daily returns
daily_returns = data.pct_change()

# Calculate volatility
volatility = daily_returns.std()
print("Daily Volatility:")
print(volatility)

# Plot volatility
plt.figure(figure=(12, 6))
sns.barplot(x=volatility.index, y=volatility)
plt.title('Daily Volatility of Stocks')
plt.xlabel('Ticker')
plt.ylabel('Volatility')
plt.show()
```


Feature Engineering

```
# Create lagged features for each stock
for ticker in tickers:
    data['Lag_1_' + ticker] = data['Adj Close', ticker].shift(1)
    data['Lag_2_' + ticker] = data['Adj Close', ticker].shift(2)

# Calculate percentage change features for each stock
for ticker in tickers:
    data['Pct_Change_' + ticker] = data['Adj Close', ticker].pct_change()

data.head()
```

| Price | Date | Adj Close, ARCO | Adj Close, GAPL | Adj Close, GEOS | Adj Close, GNWW | Adj Close, MEDP | Adj Close, PFGC | Adj Close, TEXT | Adj Close, TLX | Adj Close, UFI | Close, ARCO | Close, GAPL | Close, GEOS | Close, GNWW | Close, MEDP | Close, PFGC | Close, TEXT | Close, TLX | Close, UFI | High, ARCO | High, GAPL | High, GEOS | High, GNWW | High, MEDP | High, PFGC | High, TEXT | High, TLX | High, UFI | Low, ARCO | Low, GAPL | Low, GEOS | Low, GNWW | Low, MEDP | Low, PFGC | Low, TEXT | Low, TLX | Low, UFI | Open, ARCO | Open, GAPL | Open, GEOS | Open, GNWW | Open, MEDP | Open, PFGC | Open, TEXT | Open, TLX | Open, UFI | Volume, ARCO | Volume, GAPL | Volume, GEOS | Volume, GNWW | Volume, MEDP | Volume, PFGC | Volume, TEXT | Volume, TLX | Volume, UFI | MA30_ARCO | MA90_ARCO | MA30_GAPL | MA90_GAPL | MA30_GEOS | MA90_GEOS | MA30_GNWW | MA90_GNWW | MA30_MEDP | MA90_MEDP | MA30_PFGC | MA90_PFGC | MA30_TEXT | MA90_TEXT | MA30_TLX | MA90_TLX | MA30_UFI | MA90_UFI | MA30_CPF | MA90_CPF | Lag_1_ARCO | Lag_1_GAPL | Lag_1_GEOS | Lag_1_GNWW | Lag_1_MEDP | Lag_1_PFGC | Lag_1_TEXT | Lag_1_TLX | Lag_1_UFI | Lag_2_ARCO | Lag_2_GAPL | Lag_2_GEOS | Lag_2_GNWW | Lag_2_MEDP | Lag_2_PFGC | Lag_2_TEXT | Lag_2_TLX | Lag_2_UFI | Pct_Change_ARCO | Pct_Change_GAPL | Pct_Change_GEOS | Pct_Change_GNWW | Pct_Change_MEDP | Pct_Change_PFGC | Pct_Change_TEXT | Pct_Change_TLX | Pct_Change_UFI |
|---|---|---