**Algorithmic Trading using Python**

Algorithmic Trading is the use of algorithms in the financial market to make trading decisions. [**JP Morgan**](https://thecleverprogrammer.com/2023/01/05/heres-how-jp-morgan-uses-data-science/) Chase & Co. is one of the businesses that use Algorithmic Trading for investment decisions.

## **What is Algorithmic Trading?**

Algorithmic Trading means using algorithms in buying and selling decisions in the financial market. In an algorithmic trading strategy, a set of predefined rules are used to determine when to buy a financial instrument and when to sell it.

In simple words, Algorithmic Trading is a way of buying and selling automatically and efficiently, which is always better than trading manually.

## **Algorithmic Trading using Python**

In this section, we will implement an Algorithm Trading strategy known as the [**momentum strategy**](https://en.wikipedia.org/wiki/Momentum_investing) on stock price data using Python. In the momentum strategy, we buy the stocks when the momentum is positive and sell the stocks when the momentum is negative.

So let’s import the necessary Python libraries and collect the stock price data of Apple using the **[yfinance API](https://thecleverprogrammer.com/2021/12/21/get-stock-price-data-using-python/" \t "_blank)**:

import pandas as pd

import plotly.graph\_objs as go

from plotly.subplots import make\_subplots

import plotly.express as px

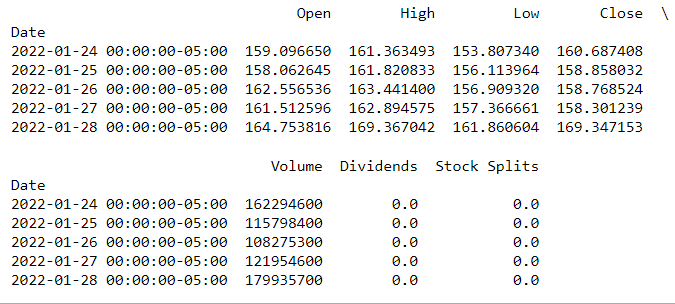
import yfinance as yf

# Get Apple's stock data from yahoo finance

stock = yf.Ticker("AAPL")

data = stock.history(period="1y")

print(data.head())



Now let’s implement the momentum strategy in Algorithmic Trading using Python:

# Calculation of momentum

data['momentum'] = data['Close'].pct\_change()

# Creating subplots to show momentum and buying/selling markers

figure = make\_subplots(rows=2, cols=1)

figure.add\_trace(go.Scatter(x=data.index,

                         y=data['Close'],

                         name='Close Price'))

figure.add\_trace(go.Scatter(x=data.index,

                         y=data['momentum'],

                         name='Momentum',

                         yaxis='y2'))

# Adding the buy and sell signals

figure.add\_trace(go.Scatter(x=data.loc[data['momentum'] > 0].index,

                         y=data.loc[data['momentum'] > 0]['Close'],

                         mode='markers', name='Buy',

                         marker=dict(color='green', symbol='triangle-up')))

figure.add\_trace(go.Scatter(x=data.loc[data['momentum'] < 0].index,

                         y=data.loc[data['momentum'] < 0]['Close'],

                         mode='markers', name='Sell',

                         marker=dict(color='red', symbol='triangle-down')))

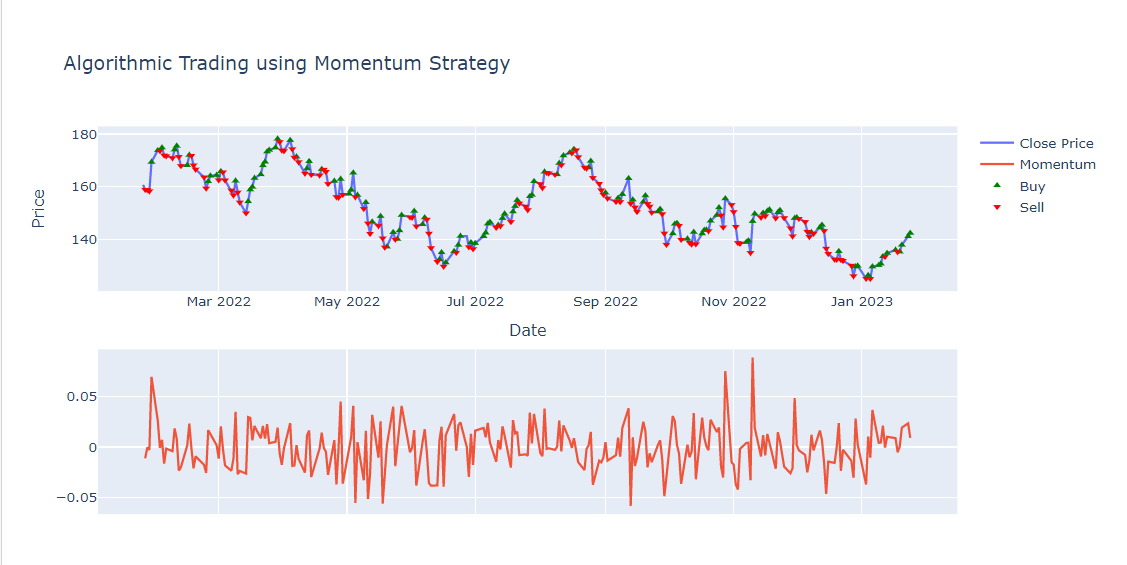
figure.update\_layout(title='Algorithmic Trading using Momentum Strategy',

                  xaxis\_title='Date',

                  yaxis\_title='Price')

figure.update\_yaxes(title="Momentum", secondary\_y=True)

figure.show()



So this is how we can implement an Algorithmic Trading strategy using the momentum strategy. In the above graph, the buy and sell signals are indicated by green triangle-up and red triangle-down markers respectively.

### **Summary**

Algorithmic Trading means using algorithms in buying and selling decisions in the financial market. In an algorithmic trading strategy, a set of predefined rules are used to determine when to buy a financial instrument and when to sell it.