Final Project

Outlook of the US economy, Stock Predictions, Trading Strategy and Automated Bot

Carlos Muñoz Fresco

Contents

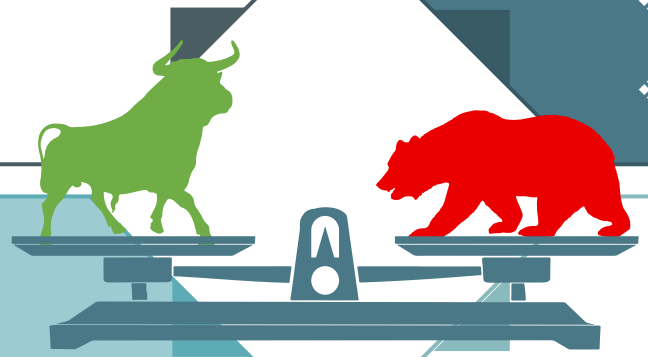
Variables

Credit Cycle

- 10 Year Bond minus 2 Year Bond Spread
- Interest Rates
- Dollar index strength (DXY)
- Volatility index (VIX)

Growth Cycle

- Gross Domestic Product (GDP) Change YoY
- Manufacturing Purchasing Managers Index (PMI)
- Unemployment Rate



Inflation Cycle

- Consumer Price Index (CPI) Change YoY

Liquidity Cycle

- Monetary Supply (M2)

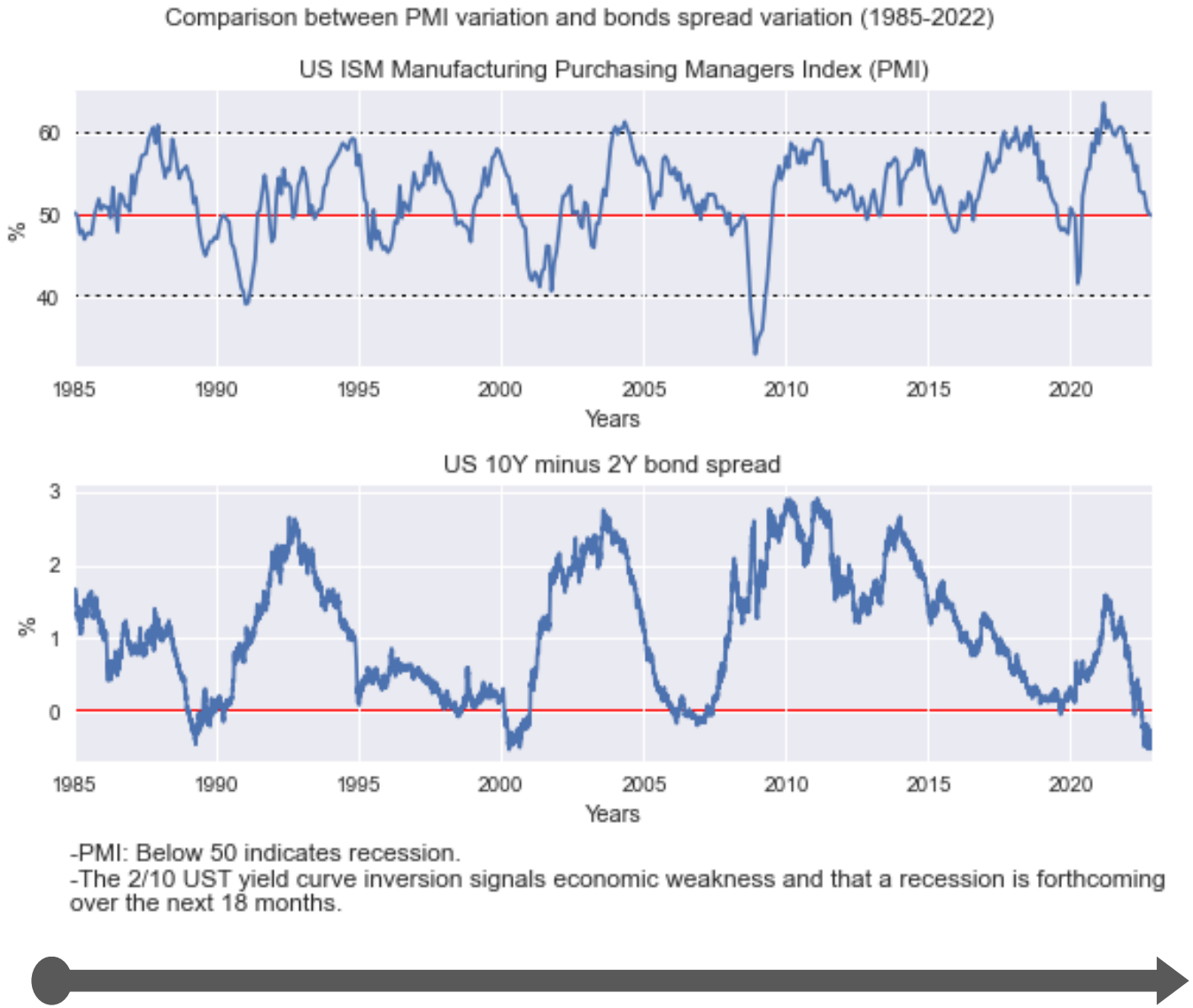
Indicators show the recession is just starting…

PMI

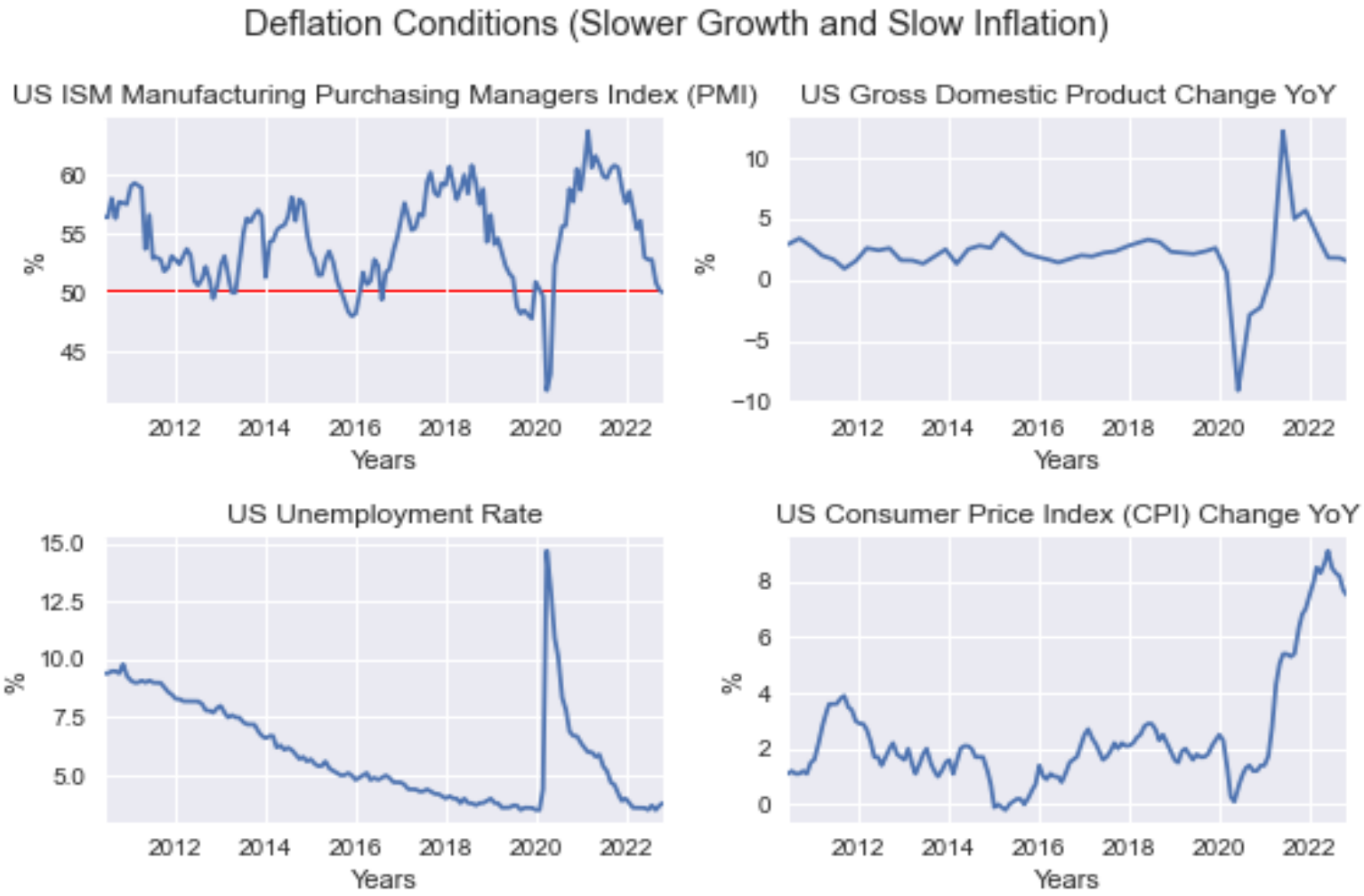
Measures the number of manufacturing activity leads that were produced in the previous month. This data is considered a very important economic and confidence measure.

10Y Bond minus 2Y Bond Spread

With increases in interest rates the value of all assets such as bonds go down. Due to this expected interest rate hike, investors who held US 2 year treasury bonds start selling them. The sell-off brings down the price of 2 year US treasury bonds. Decline in price results in increases in yield.

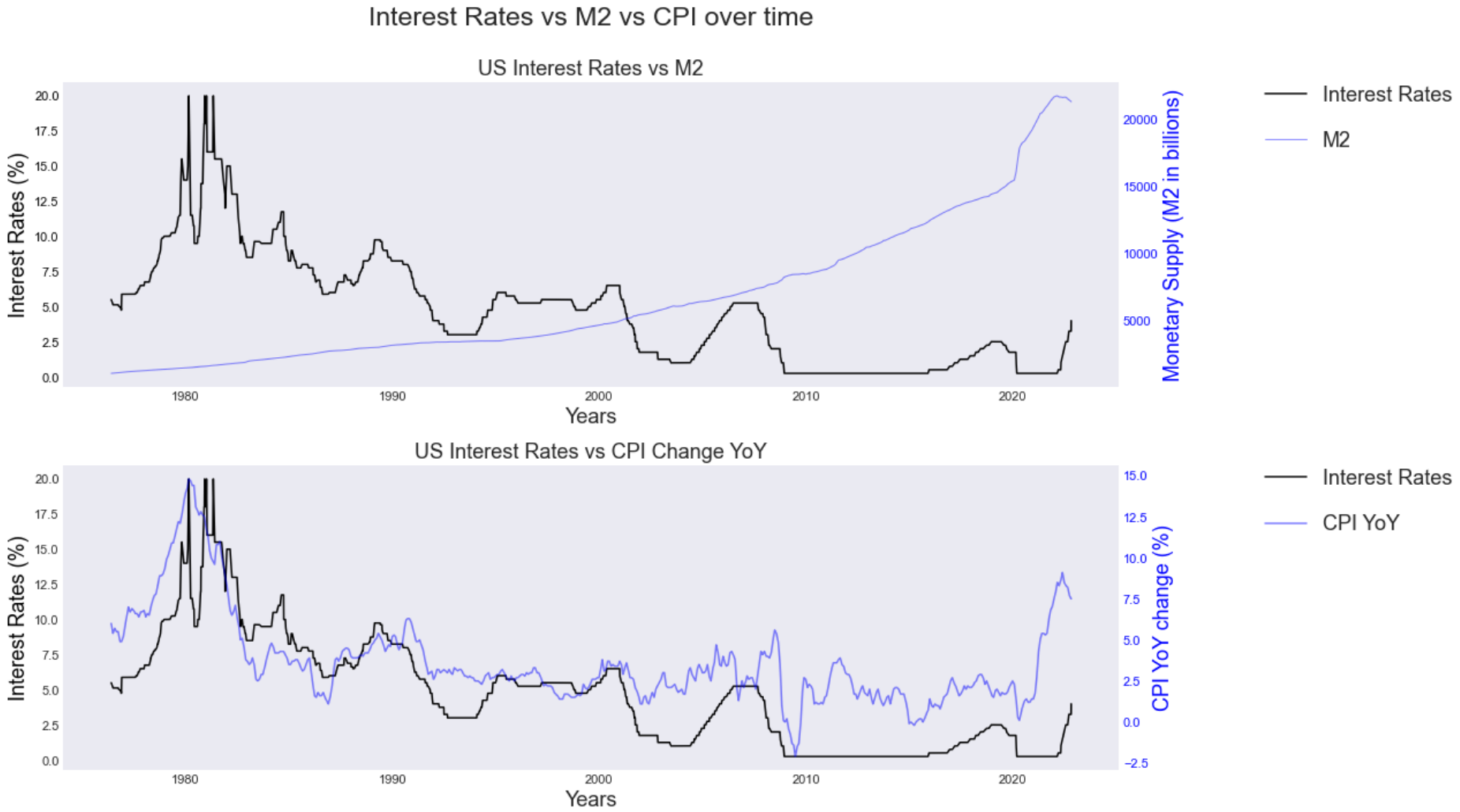


Current Market Conditions



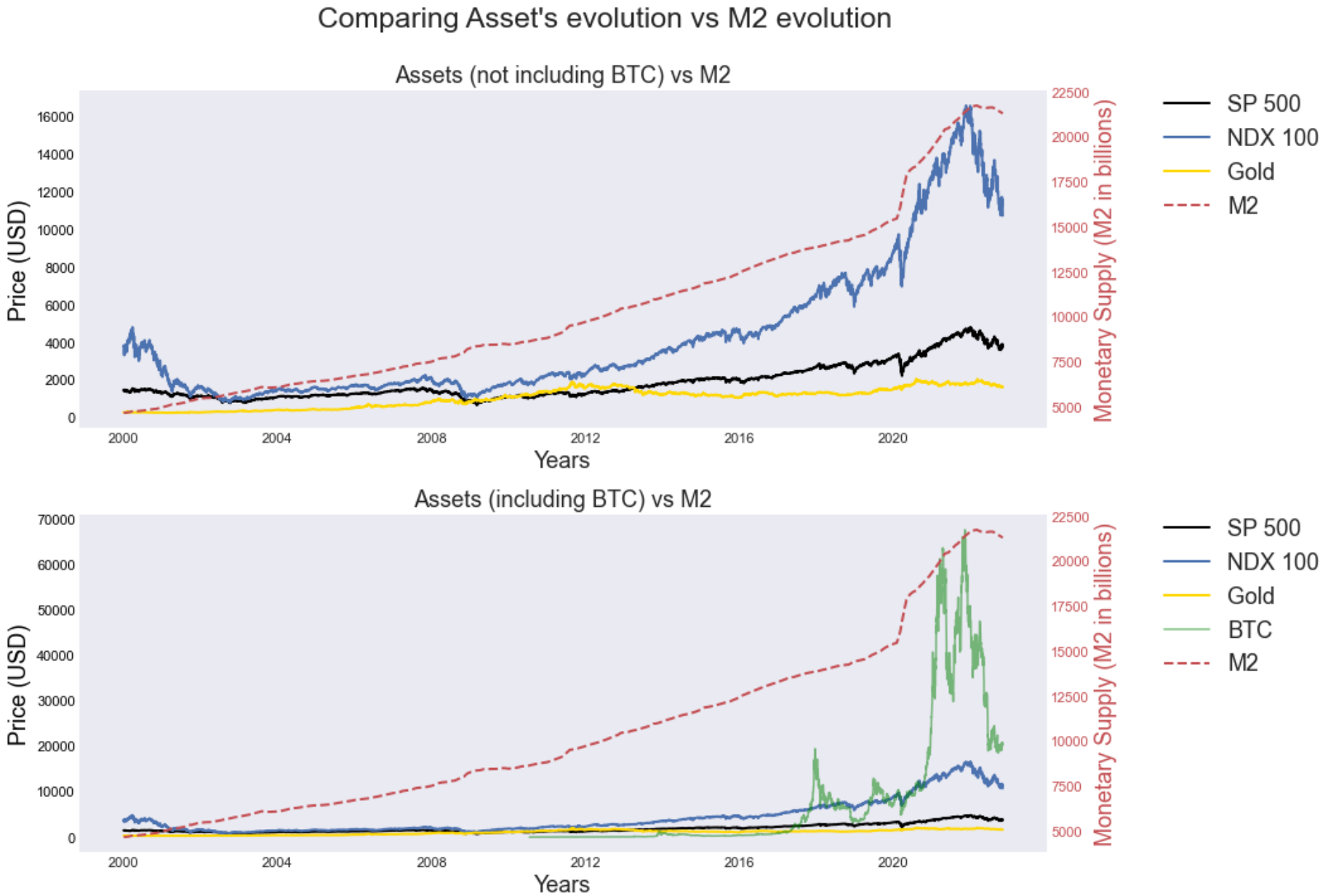
In the next months PMI, GDP and CPI are going to slow down.
Unemployment rates are going to increase as economic activity deteriorates.

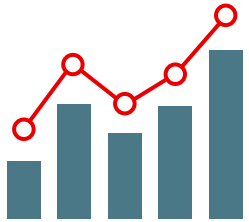
Identifying Long Term Trends



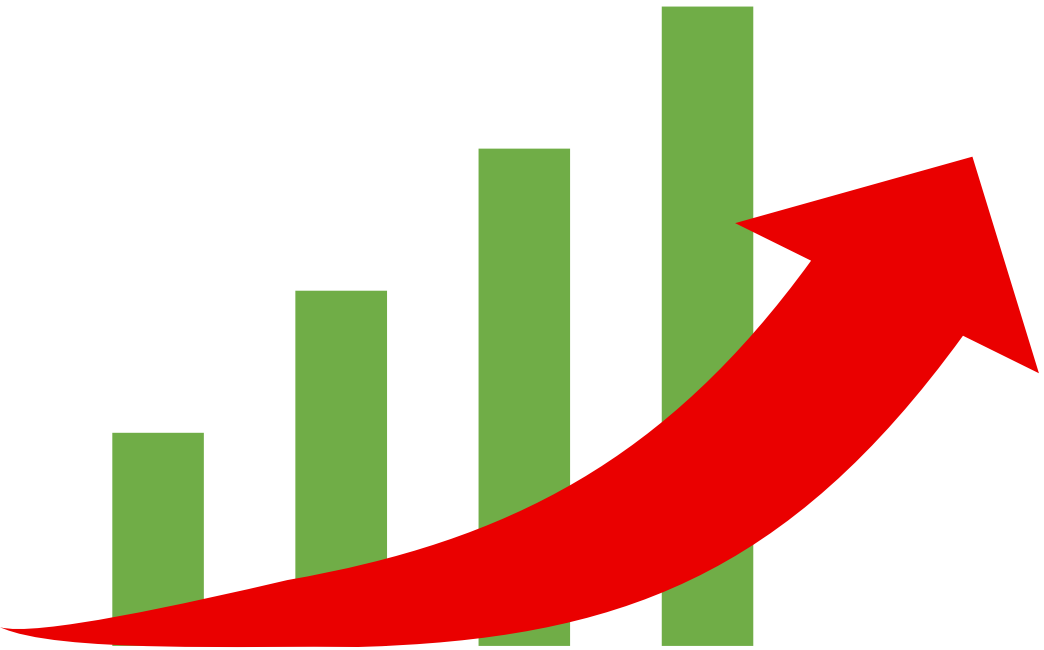
Interest Rates start to rise at times when inflation needs to be controlled.
IR : CPI --> Positive lagged correlation (IR follow CPI data).
Exponential money printing within the years.

Is money printing the main incentive for higher asset prices?





Prediction Model



Prophet()

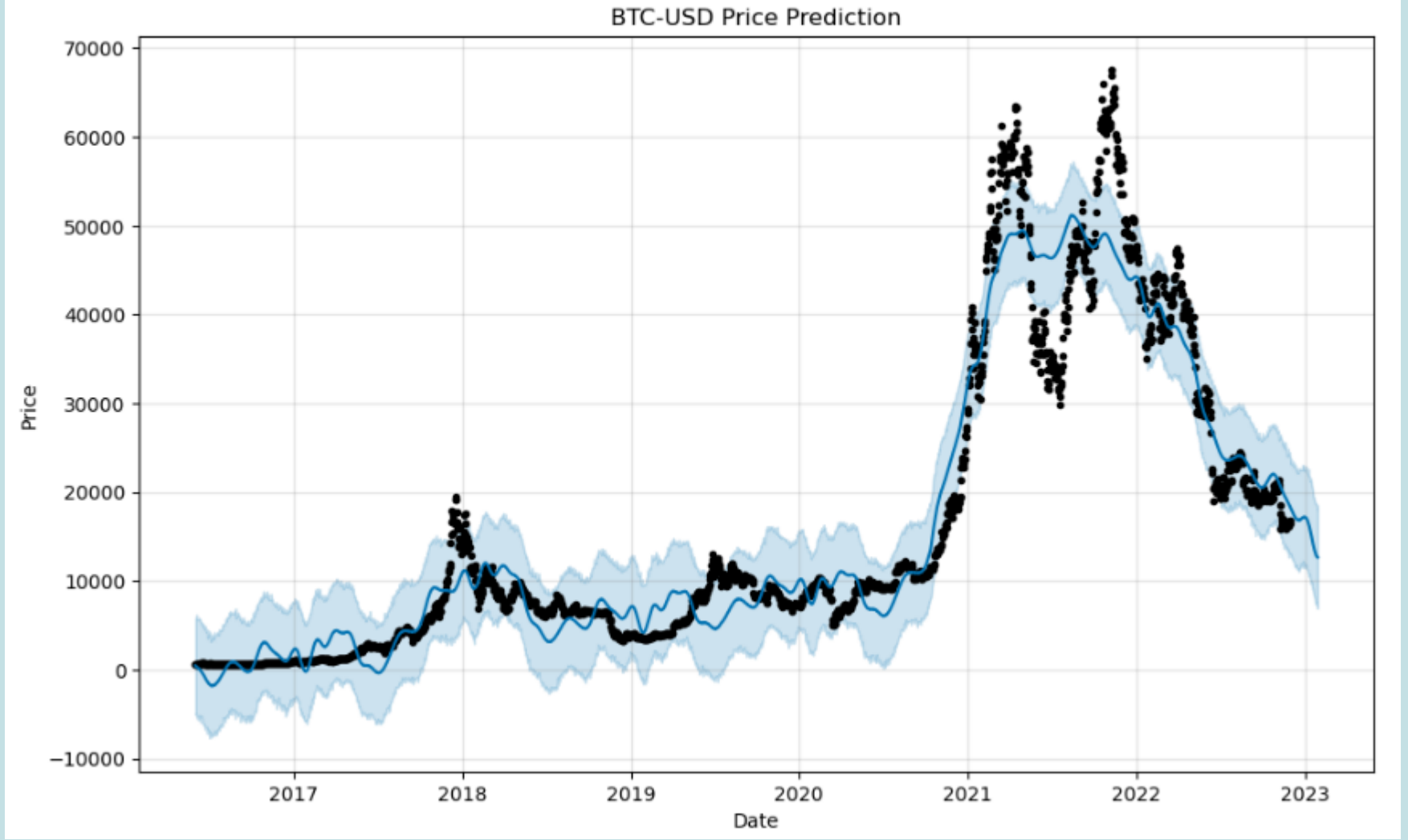
```
def price_prediction(symbol, start, days):
    df= yf.download(symbol, start)
    df = df.reset_index()
    df[["ds", "y"]] = df[["Date", "Adj Close"]]
    model = Prophet()
    model.fit(df)
    future = model.make_future_dataframe(days)
    forecast = model.predict(future)
    print(forecast["trend"][-1:])
    model.plot(forecast)
    plt.title(f'{symbol} Price Prediction')
    plt.xlabel("Date")
    plt.ylabel("Price")
    return plt.show()
```

```
price_prediction("BTC-USD", '2016-06-01', 60)
```

```
[**********100%**********]  1 of 1 completed
```

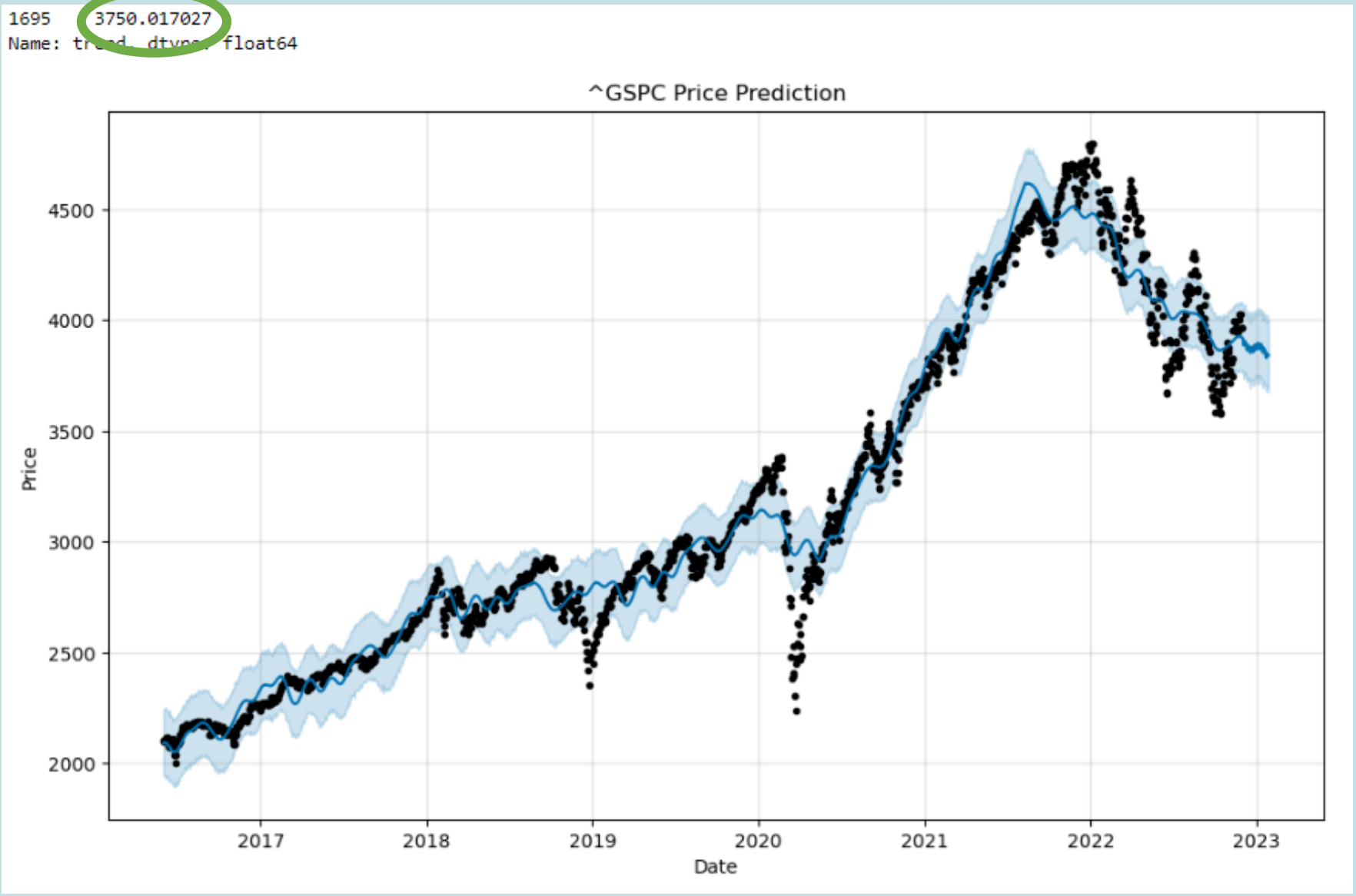
Bitcoin price prediction


```
2433    12970.549975
Name: trend, dtype: float64
```



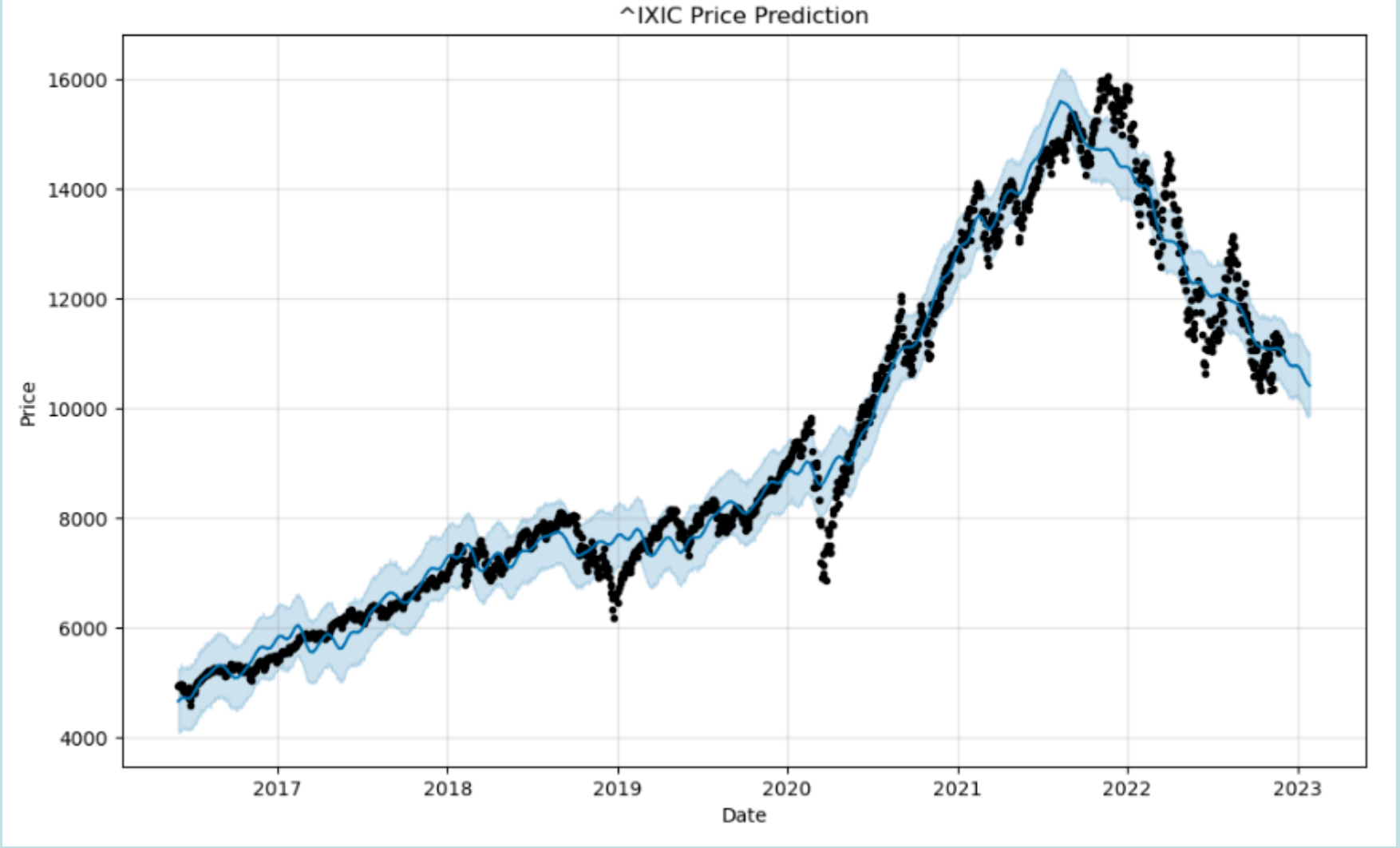
SP 500 price prediction


```
1695    3750.017027
Name: trend, dtype: float64
```



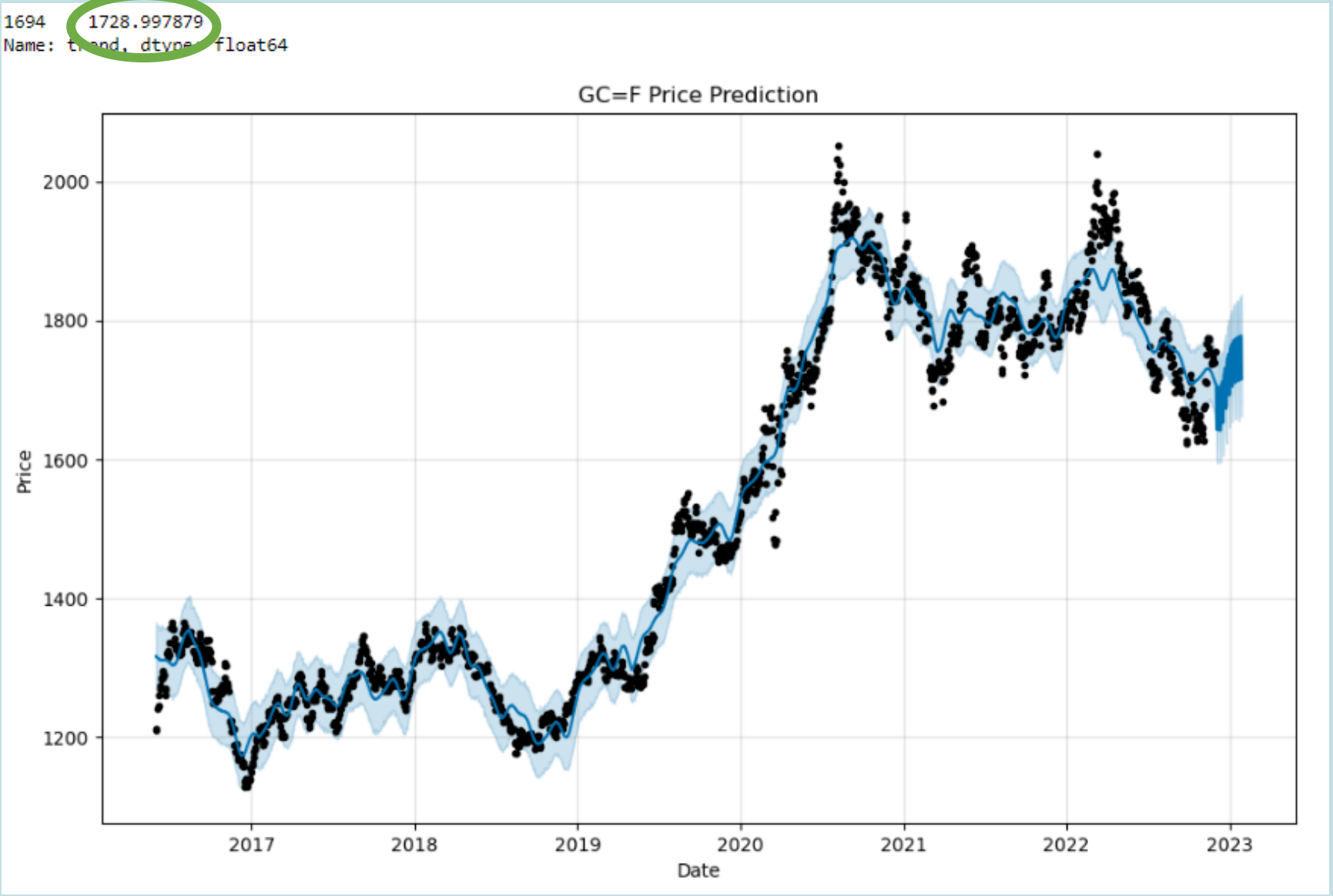
NASDAQ 100 price prediction


```
1695    10199.105709
Name: trend, dtype: float64
```



Gold price prediction


```
1694    1728.997879
Name: trend, dtype: float64
```



```python
def Buy_RSI(btc_15m):
  sigPriceBuy = []
  sigPriceSell = []
  flag = -1
  for i in range(len(btc_15m)):
    if (btc_15m ['RSI'][i] < 20 and btc_15m ['K'][i] < 20):
      if flag != 1:
        sigPriceBuy.append(btc_15m['close'][i])
        sigPriceSell.append(np.nan)
        flag = 1
      else:
        sigPriceBuy.append(np.nan)
        sigPriceSell.append(np.nan)
    elif (btc_15m ["RSI"][i] > 45):
      if flag != 0:
        sigPriceBuy.append(np.nan)
        sigPriceSell.append(btc_15m["close"][i])
        flag = 0
      else:
        sigPriceBuy.append(np.nan)
        sigPriceSell.append(np.nan)
    else:
        sigPriceBuy.append(np.nan)
        sigPriceSell.append(np.nan)
  return(sigPriceBuy, sigPriceSell)
```

```python
Buy_RSI = Buy_RSI(btc_15m)
btc_15m['Long Signal RSI'] = Buy_RSI[0]
btc_15m['Close Long Signal RSI'] = Buy_RSI[1]
# To show the data
btc_15m.head()
```

Bearish Trend > I wanted to catch small movements and get quickly out of trades.

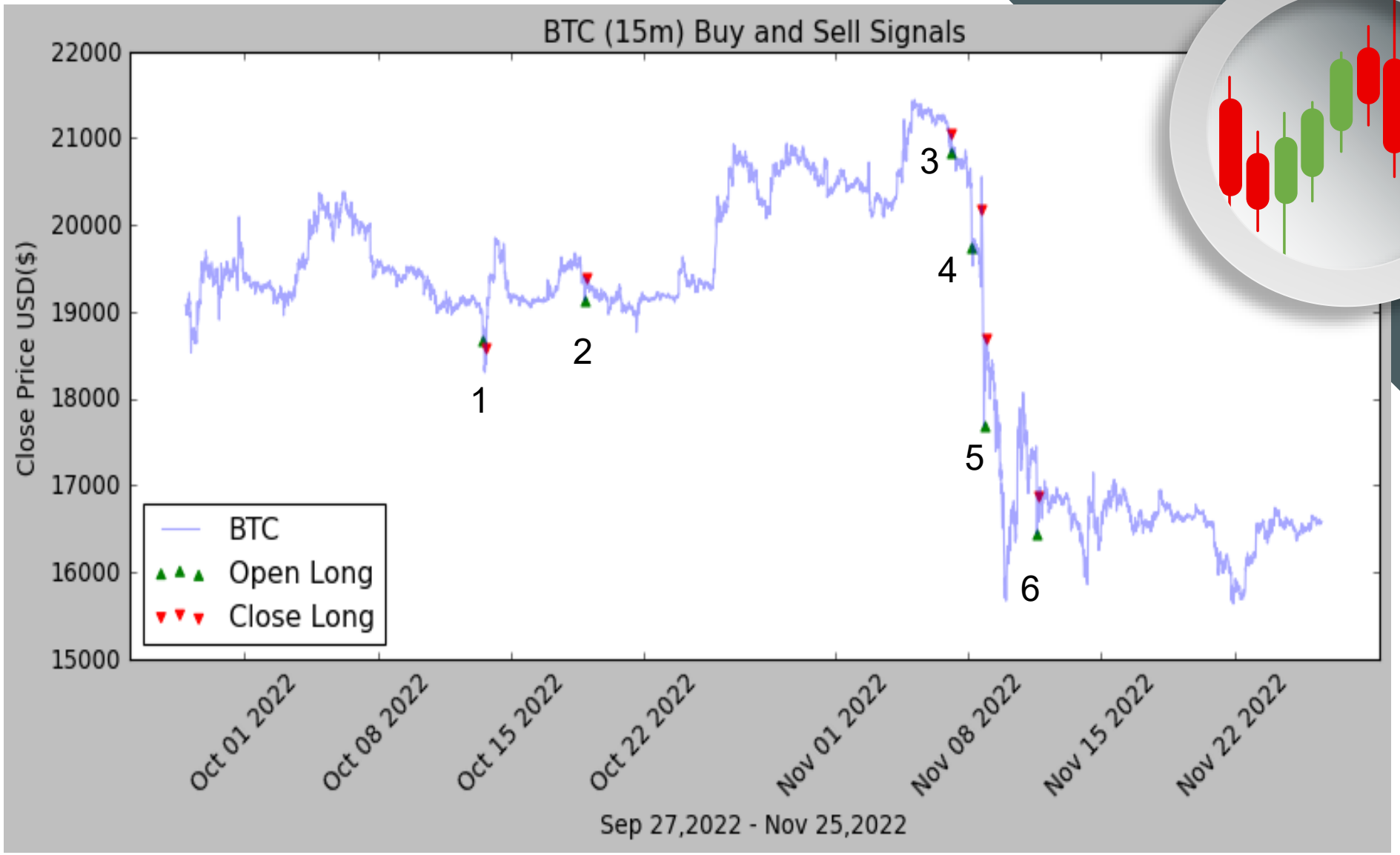
Trading Strategy & Bot

Asset: BTC (15 minutes timeframe)

Buying Conditions > **RSI < 20 (Oversold conditions)**
K < 20 (Low momentum)

Selling Conditions > **RSI > 45**

Back testing Results



Results

	Open	Close	Gained or Lost %
0	18673.5	18585.0	-0.473934
1	19126.0	19384.0	1.348949
2	20834.0	21047.0	1.022367
3	19734.5	20185.0	2.282804
4	17681.5	18695.5	5.734808
5	16437.0	16882.5	2.710349
Row_Total	112486.5	114779.0	12.625343