

Code

```
In [5]: pip install yfinance

Collecting yfinanceNote: you may need to restart the kernel to use updated packages.

[notice] A new release of pip available: 22.2.2 > 22.3
[notice] To update, run: python.exe -m pip install --upgrade pip

Downloading yfinance-0.1.80-py2.py3-none-any.whl (20 kB)
Collecting multitasking==0.0.7
  Downloading multitasking-0.0.11-py3-none-any.whl (8.5 kB)
Collecting requests>=2.26
  Downloading requests-2.28.1-py3-none-any.whl (62 kB)
----- 62.8/62.3 kB 1.7 MB/s eta 0:00:00
Requirement already satisfied: pandas>=0.24.0 in c:\users\91998\appdata\local\programs\python\python310\lib\site-packages (from yfinance) (1.4.4)
Requirement already satisfied: lxml>=4.5.1 in c:\users\91998\appdata\local\programs\python\python310\lib\site-packages (from yfinance) (4.9.1)
Requirement already satisfied: numpy>=1.15 in c:\users\91998\appdata\local\programs\python\python310\lib\site-packages (from yfinance) (1.23.3)
Collecting apcdirs>=1.4.4

In [4]: import yfinance as yf
        msft = yf.Ticker("MSFT")
        msft_hist = msft.history(period="max")
```

```
In [7]: import os
import pandas as pd

DATA_PATH = "msft_data.json"

if os.path.exists(DATA_PATH):
    with open(DATA_PATH) as f:
        msft_hist = pd.read_json(DATA_PATH)
else:
    msft = yf.Ticker("MSFT")
    msft_hist = msft.history(period="max")
    msft_hist.to_json(DATA_PATH)
```

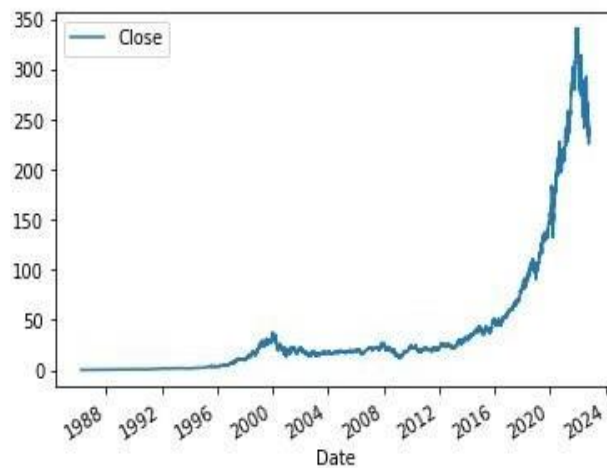
```
In [8]: msft_hist.head(5)
```

Out[8]:

	Open	High	Low	Close	Volume	Dividends	Stock Splits
Date							
1986-03-13 00:00:00-05:00	0.055536	0.063703	0.055536	0.060980	1031788800	0.0	0.0
1986-03-14 00:00:00-05:00	0.060980	0.064247	0.060980	0.063158	308160000	0.0	0.0
1986-03-17 00:00:00-05:00	0.063158	0.064792	0.063158	0.064247	133171200	0.0	0.0
1986-03-18 00:00:00-05:00	0.064247	0.064792	0.062069	0.062613	67766400	0.0	0.0
1986-03-19 00:00:00-05:00	0.062613	0.063158	0.060980	0.061524	47894400	0.0	0.0

```
In [9]: msft_hist.plot.line(y="Close", use_index=True)
```

```
Out[9]: <AxesSubplot: xlabel='Date'>
```



```
In [10]: data = msft_hist[["Close"]]
data = data.rename(columns = {'Close':'Actual_Close'})
data["Target"] = msft_hist.rolling(2).apply(lambda x: x.iloc[1] > x.iloc[0])["Close"]
```

```
In [11]: data.head()
```

```
Out[11]:
```

	Actual_Close	Target
Date		
1986-03-13 00:00:00-05:00	0.060980	NaN
1986-03-14 00:00:00-05:00	0.063158	1.0
1986-03-17 00:00:00-05:00	0.064247	1.0
1986-03-18 00:00:00-05:00	0.062613	0.0
1986-03-19 00:00:00-05:00	0.061524	0.0

```
In [12]: msft_prev = msft_hist.copy()
msft_prev = msft_prev.shift(1)
```

```
In [13]: msft_prev.head()
```

```
Out[13]:
```

	Open	High	Low	Close	Volume	Dividends	Stock Splits
Date							
1986-03-13 00:00:00-05:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1986-03-14 00:00:00-05:00	0.055536	0.063703	0.055536	0.060980	1.031789e+09	0.0	0.0
1986-03-17 00:00:00-05:00	0.060980	0.064247	0.060980	0.063158	3.081600e+08	0.0	0.0
1986-03-18 00:00:00-05:00	0.063158	0.064792	0.063158	0.064247	1.331712e+08	0.0	0.0
1986-03-19 00:00:00-05:00	0.064247	0.064792	0.062069	0.062613	6.776640e+07	0.0	0.0

```
In [14]: predictors = ["Close", "Volume", "Open", "High", "Low"]  
data = data.join(msft_prev[predictors]).iloc[1:]
```

```
In [15]: data.head()
```

```
Out[15]:
```

	Actual_Close	Target	Close	Volume	Open	High	Low
Date							
1986-03-14 00:00:00-05:00	0.063158	1.0	0.060980	1.031789e+09	0.055536	0.063703	0.055536
1986-03-17 00:00:00-05:00	0.064247	1.0	0.063158	3.081600e+08	0.060980	0.064247	0.060980
1986-03-18 00:00:00-05:00	0.062613	0.0	0.064247	1.331712e+08	0.063158	0.064792	0.063158
1986-03-19 00:00:00-05:00	0.061524	0.0	0.062613	6.776640e+07	0.064247	0.064792	0.062069
1986-03-20 00:00:00-05:00	0.059891	0.0	0.061524	4.789440e+07	0.062613	0.063158	0.060980

```
In [16]: from sklearn.ensemble import RandomForestClassifier  
import numpy as np  
model = RandomForestClassifier(n_estimators=100, min_samples_split=200, random_state=1)
```

```
In [17]: train = data.iloc[:-100]  
test = data.iloc[-100:]  
  
model.fit(train[predictors], train["Target"])
```

```
Out[17]:
```

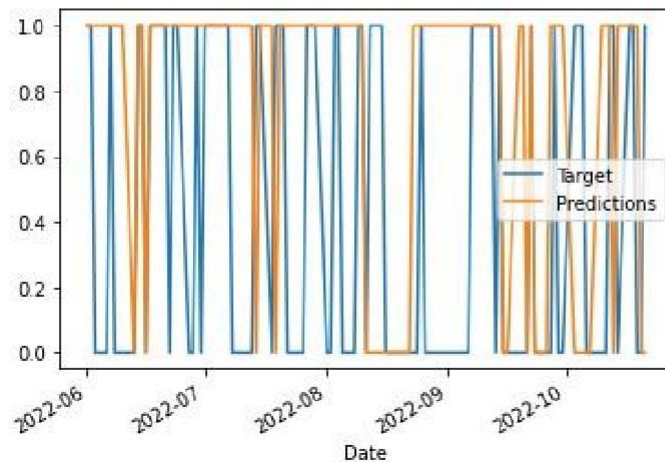
```
RandomForestClassifier  
RandomForestClassifier(min_samples_split=200, random_state=1)
```

```
In [18]: from sklearn.metrics import precision_score
preds = model.predict(test[predictors])
preds = pd.Series(preds, index=test.index)
precision_score(test["Target"], preds)
```

Out[18]: 0.4666666666666667

```
In [19]: combined = pd.concat({"Target": test["Target"], "Predictions": preds}, axis=1)
combined.plot()
```

Out[19]: <AxesSubplot: xlabel='Date'>



RESULT

