In [31]:
```
import yfinance as yf
import pandas as pd
import os
```

In [32]:
```
if os.path.exists("sp500.csv"):
    sp500 = pd.read_csv("sp500.csv", index_col=0)
else:
    sp500 = yf.Ticker("^GSPC")
    sp500 = sp500.history(period="max")
    sp500.to_csv("sp500.csv")
```

In [33]:
```
sp500.index = pd.to_datetime(sp500.index)
```

In [34]:
```
sp500
```

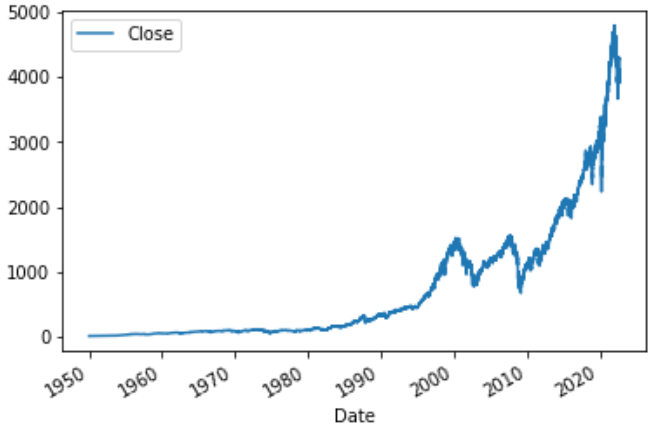
Out[34]:

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
1950-01-03	16.660000	16.660000	16.660000	16.660000	1260000	0	0
1950-01-04	16.850000	16.850000	16.850000	16.850000	1890000	0	0
1950-01-05	16.930000	16.930000	16.930000	16.930000	2550000	0	0
1950-01-06	16.980000	16.980000	16.980000	16.980000	2010000	0	0
1950-01-09	17.080000	17.080000	17.080000	17.080000	2520000	0	0
...	...	...	...	...	...	...	...
2022-09-06	3930.889893	3942.550049	3886.750000	3908.189941	2209800080	0	0
2022-09-07	3909.429932	3987.889893	3906.030029	3979.870117	0	0	0
2022-09-08	3959.939941	4010.500000	3944.810059	4006.179932	0	0	0
2022-09-09	4022.939941	4076.810059	4022.939941	4067.360107	0	0	0
2022-09-12	4083.669922	4119.279785	4083.669922	4107.279785	1602969000	0	0

18292 rows × 7 columns

In [35]:
```
sp500.plot.line(y="Close", use_index=True)
```

Out[35]:
```
<AxesSubplot:xlabel='Date'>
```



In [36]:
```
del sp500["Dividends"]
del sp500["Stock Splits"]
```

In [37]:
```
sp500["Tomorrow"] = sp500["Close"].shift(-1)
```

In [38]:
```
sp500["Target"] = (sp500["Tomorrow"] > sp500["Close"]).astype(int)
```

In [39]:
```
sp500 = sp500.loc["1990-01-01":].copy()
```

In [40]:
```
sp500
```

Out[40]:

Date	Open	High	Low	Close	Volume	Tomorrow	Target
1990-01-02	353.399994	359.690002	351.980011	359.690002	162070000	358.760010	0
1990-01-03	359.690002	360.589996	357.890015	358.760010	192330000	355.670013	0
1990-01-04	358.760010	358.760010	352.890015	355.670013	177000000	352.200012	0
1990-01-05	355.670013	355.670013	351.350006	352.200012	158530000	353.790009	1
1990-01-08	352.200012	354.239990	350.540009	353.790009	140110000	349.619995	0
...	...	...	...	...	...	...	...
2022-09-06	3930.889893	3942.550049	3886.750000	3908.189941	2209800080	3979.870117	1
2022-09-07	3909.429932	3987.889893	3906.030029	3979.870117	0	4006.179932	1
2022-09-08	3959.939941	4010.500000	3944.810059	4006.179932	0	4067.360107	1
2022-09-09	4022.939941	4076.810059	4022.939941	4067.360107	0	4107.279785	1
2022-09-12	4083.669922	4119.279785	4083.669922	4107.279785	1602969000	NaN	0

8238 rows × 7 columns

In [41]:
```
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(n_estimators=100, min_samples_split=100, random_state=1)

train = sp500.iloc[:-100]
test = sp500.iloc[-100:]

predictors = ["Close", "Volume", "Open", "High", "Low"]
model.fit(train[predictors], train["Target"])
```

Out[41]:
```
RandomForestClassifier(min_samples_split=100, random_state=1)
```

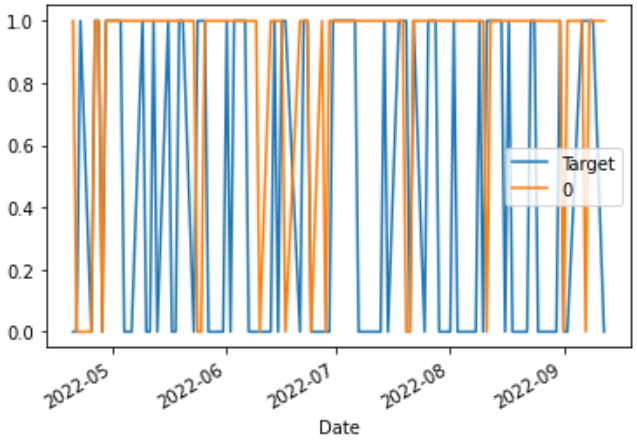
In [42]:
```
from sklearn.metrics import precision_score

preds = model.predict(test[predictors])
preds = pd.Series(preds, index=test.index)
precision_score(test["Target"], preds)
```

Out[42]:
```
0.47058823529411764
```

In [43]:
```
combined = pd.concat([test["Target"], preds], axis=1)
combined.plot()
```

Out[43]: <AxesSubplot:xlabel='Date'>



In [44]:
```
def predict(train, test, predictors, model):
    model.fit(train[predictors], train["Target"])
    preds = model.predict(test[predictors])
    preds = pd.Series(preds, index=test.index, name="Predictions")
    combined = pd.concat([test["Target"], preds], axis=1)
    return combined
```

In [45]:
```
def backtest(data, model, predictors, start=2500, step=250):
    all_predictions = []

    for i in range(start, data.shape[0], step):
        train = data.iloc[0:i].copy()
        test = data.iloc[i:(i+step)].copy()
        predictions = predict(train, test, predictors, model)
        all_predictions.append(predictions)

    return pd.concat(all_predictions)
```

In [46]:
```
predictions = backtest(sp500, model, predictors)
```

In [47]:
```
predictions["Predictions"].value_counts()
```

Out[47]:
```
0    3337
1    2401
Name: Predictions, dtype: int64
```

In [48]:
```
precision_score(predictions["Target"], predictions["Predictions"])
```

Out[48]: 0.534777176176593

In [49]:
```
predictions["Target"].value_counts() / predictions.shape[0]
```

Out[49]:
```
1    0.536075
0    0.463925
Name: Target, dtype: float64
```

In [50]:
```
horizons = [2,5,60,250,1000]
new_predictors = []

for horizon in horizons:
    rolling_averages = sp500.rolling(horizon).mean()

    ratio_column = f"Close_Ratio_{horizon}"
    sp500[ratio_column] = sp500["Close"] / rolling_averages["Close"]

    trend_column = f"Trend_{horizon}"
    sp500[trend_column] = sp500.shift(1).rolling(horizon).sum()["Target"]

    new_predictors+= [ratio_column, trend_column]
```

In [53]:
```
sp500 = sp500.dropna(subset=sp500.columns[sp500.columns != "Tomorrow"])
```

In [55]:
```
sp500
```

Out[55]:

Date	Open	High	Low	Close	Volume	Tomorrow	Target	Close_Ratio_2	Trend
1993-12-14	465.730011	466.119995	462.459991	463.059998	275050000	461.839996	0	0.997157	
1993-12-15	463.059998	463.690002	461.839996	461.839996	331770000	463.339996	1	0.998681	
1993-12-16	461.859985	463.980011	461.859985	463.339996	284620000	466.380005	1	1.001621	
1993-12-17	463.339996	466.380005	463.339996	466.380005	363750000	465.850006	0	1.003270	
1993-12-20	466.380005	466.899994	465.529999	465.850006	255900000	465.299988	0	0.999431	
...	...	...	...	...	...	...	...	...	
2022-09-06	3930.889893	3942.550049	3886.750000	3908.189941	2209800080	3979.870117	1	0.997948	
2022-09-07	3909.429932	3987.889893	3906.030029	3979.870117	0	4006.179932	1	1.009087	
2022-09-08	3959.939941	4010.500000	3944.810059	4006.179932	0	4067.360107	1	1.003294	
2022-09-09	4022.939941	4076.810059	4022.939941	4067.360107	0	4107.279785	1	1.007578	
2022-09-12	4083.669922	4119.279785	4083.669922	4107.279785	1602969000	NaN	0	1.004883	

7238 rows × 17 columns

In [56]:
```
model = RandomForestClassifier(n_estimators=200, min_samples_split=50, random_state=1)
```

In [57]:
```
def predict(train, test, predictors, model):
    model.fit(train[predictors], train["Target"])
    preds = model.predict_proba(test[predictors])[:,1]
    preds[preds >=.6] = 1
    preds[preds <.6] = 0
    preds = pd.Series(preds, index=test.index, name="Predictions")
    combined = pd.concat([test["Target"], preds], axis=1)
    return combined
```

In [58]:
```
predictions = backtest(sp500, model, new_predictors)
```

In [59]:
```
predictions["Predictions"].value_counts()
```

Out[59]:
```
0.0    3933
1.0     805
Name: Predictions, dtype: int64
```

In [60]:
```
precision_score(predictions["Target"], predictions["Predictions"])
```

Out[60]:
```
0.5701863354037268
```

In [61]:
```
predictions["Target"].value_counts() / predictions.shape[0]
```

Out[61]:
```
1    0.546855
0    0.453145
Name: Target, dtype: float64
```

In [62]:
```
predictions
```

Out[62]:

Date	Target	Predictions
2003-11-14	0	0.0
2003-11-17	0	1.0
2003-11-18	1	1.0
2003-11-19	0	0.0
2003-11-20	1	1.0
...	...	...
2022-09-06	1	0.0
2022-09-07	1	0.0
2022-09-08	1	0.0
2022-09-09	1	0.0
2022-09-12	0	0.0

4738 rows × 2 columns

In [ ]:
```

```