trading model

October 28, 2023

[3]:
```
!pip install yfinance
```

```
Collecting yfinance
  Downloading yfinance-0.2.31-py2.py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: pandas>=1.3.0 in
c:\users\morea\anaconda3\lib\site-packages (from yfinance) (1.5.3)
Requirement already satisfied: numpy>=1.16.5 in
c:\users\morea\anaconda3\lib\site-packages (from yfinance) (1.24.3)
Collecting requests>=2.31 (from yfinance)
  Downloading requests-2.31.0-py3-none-any.whl.metadata (4.6 kB)
Collecting multitasking>=0.0.7 (from yfinance)
  Downloading multitasking-0.0.11-py3-none-any.whl (8.5 kB)
Requirement already satisfied: lxml>=4.9.1 in c:\users\morea\anaconda3\lib\site-
packages (from yfinance) (4.9.2)
Requirement already satisfied: appdirs>=1.4.4 in
c:\users\morea\anaconda3\lib\site-packages (from yfinance) (1.4.4)
Requirement already satisfied: pytz>=2022.5 in
c:\users\morea\anaconda3\lib\site-packages (from yfinance) (2022.7)
Collecting frozendict>=2.3.4 (from yfinance)
  Downloading frozendict-2.3.8-py311-none-any.whl (14 kB)
Collecting peewee>=3.16.2 (from yfinance)
  Downloading peewee-3.17.0.tar.gz (2.9 MB)
     ---------------------------------------- 0.0/2.9 MB ? eta -:--:--
     ---------------------------------------- 0.0/2.9 MB 1.3 MB/s eta 0:00:03
     - -------------------------------------- 0.1/2.9 MB 907.3 kB/s eta 0:00:04
     - -------------------------------------- 0.1/2.9 MB 1.2 MB/s eta 0:00:03
     --- ------------------------------------ 0.2/2.9 MB 1.5 MB/s eta 0:00:02
     ------ --------------------------------- 0.5/2.9 MB 2.2 MB/s eta 0:00:02
     ------- -------------------------------- 0.5/2.9 MB 2.2 MB/s eta 0:00:02
     ------- -------------------------------- 0.6/2.9 MB 1.9 MB/s eta 0:00:02
     ----------- ---------------------------- 0.9/2.9 MB 2.5 MB/s eta 0:00:01
     ------------- -------------------------- 1.0/2.9 MB 2.6 MB/s eta 0:00:01
     -------------- ------------------------- 1.0/2.9 MB 2.5 MB/s eta 0:00:01
     ----------------- ---------------------- 1.3/2.9 MB 2.7 MB/s eta 0:00:01
     ------------------------ --------------- 1.8/2.9 MB 3.3 MB/s eta 0:00:01
     --------------------------- ------------ 2.0/2.9 MB 3.5 MB/s eta 0:00:01
     --------------------------- ------------ 2.0/2.9 MB 3.5 MB/s eta 0:00:01
     ------------------------------------ ---- 2.6/2.9 MB 3.9 MB/s eta 0:00:01
```

```
     ---------------------------------------- 2.9/2.9 MB 4.2 MB/s eta 0:00:00
  Installing build dependencies: started
  Installing build dependencies: finished with status 'done'
  Getting requirements to build wheel: started
  Getting requirements to build wheel: finished with status 'done'
  Preparing metadata (pyproject.toml): started
  Preparing metadata (pyproject.toml): finished with status 'done'
Requirement already satisfied: beautifulsoup4>=4.11.1 in
c:\users\morea\anaconda3\lib\site-packages (from yfinance) (4.12.2)
Collecting html5lib>=1.1 (from yfinance)
  Downloading html5lib-1.1-py2.py3-none-any.whl (112 kB)
     ---------------------------------------- 0.0/112.2 kB ? eta -:--:--
     -------------------------------------- 112.2/112.2 kB 6.8 MB/s eta 0:00:00
Requirement already satisfied: soupsieve>1.2 in
c:\users\morea\anaconda3\lib\site-packages (from
beautifulsoup4>=4.11.1->yfinance) (2.4)
Requirement already satisfied: six>=1.9 in c:\users\morea\anaconda3\lib\site-
packages (from html5lib>=1.1->yfinance) (1.16.0)
Requirement already satisfied: webencodings in
c:\users\morea\anaconda3\lib\site-packages (from html5lib>=1.1->yfinance)
(0.5.1)
Requirement already satisfied: python-dateutil>=2.8.1 in
c:\users\morea\anaconda3\lib\site-packages (from pandas>=1.3.0->yfinance)
(2.8.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
c:\users\morea\anaconda3\lib\site-packages (from requests>=2.31->yfinance)
(2.0.4)
Requirement already satisfied: idna<4,>=2.5 in
c:\users\morea\anaconda3\lib\site-packages (from requests>=2.31->yfinance) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in
c:\users\morea\anaconda3\lib\site-packages (from requests>=2.31->yfinance)
(1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in
c:\users\morea\anaconda3\lib\site-packages (from requests>=2.31->yfinance)
(2023.7.22)
Downloading yfinance-0.2.31-py2.py3-none-any.whl (65 kB)
   ---------------------------------------- 0.0/65.6 kB ? eta -:--:--
   ---------------------------------------- 65.6/65.6 kB 3.7 MB/s eta 0:00:00
Downloading requests-2.31.0-py3-none-any.whl (62 kB)
   ---------------------------------------- 0.0/62.6 kB ? eta -:--:--
   ---------------------------------------- 62.6/62.6 kB ? eta 0:00:00
Building wheels for collected packages: peewee
  Building wheel for peewee (pyproject.toml): started
  Building wheel for peewee (pyproject.toml): finished with status 'done'
  Created wheel for peewee: filename=peewee-3.17.0-py3-none-any.whl size=135767
sha256=1393587bd0e64add0efbfb31417a0bdefb8f5f63596c36e47a30f213623015a3
  Stored in directory: c:\users\morea\appdata\local\pip\cache\wheels\02\20\23\74
a10d0cd31f5d41c19b92ddf4c138ceff01b9f4675f19dbf5
```

```
Successfully built peewee
Installing collected packages: peewee, multitasking, requests, html5lib,
frozendict, yfinance
  Attempting uninstall: requests
    Found existing installation: requests 2.29.0
    Uninstalling requests-2.29.0:
      Successfully uninstalled requests-2.29.0
Successfully installed frozendict-2.3.8 html5lib-1.1 multitasking-0.0.11
peewee-3.17.0 requests-2.31.0 yfinance-0.2.31
```

```
ERROR: pip's dependency resolver does not currently take into account all the
packages that are installed. This behaviour is the source of the following
dependency conflicts.
conda-repo-cli 1.0.41 requires requests_mock, which is not installed.
transformers 2.1.1 requires sentencepiece, which is not installed.
conda-repo-cli 1.0.41 requires clyent==1.2.1, but you have clyent 1.2.2 which is
incompatible.
conda-repo-cli 1.0.41 requires nbformat==5.4.0, but you have nbformat 5.7.0
which is incompatible.
conda-repo-cli 1.0.41 requires requests==2.28.1, but you have requests 2.31.0
which is incompatible.
```

[4]:
```python
import yfinance as yf
```

[5]:
```python
sp500=yf.Ticker("^GSPC")
```

[6]:
```python
sp500=sp500.history(period='max')
```

[7]:
```python
sp500
```

[7]:
```
                                  Open         High          Low        Close  \
Date
1927-12-30 00:00:00-05:00    17.660000    17.660000    17.660000    17.660000
1928-01-03 00:00:00-05:00    17.760000    17.760000    17.760000    17.760000
1928-01-04 00:00:00-05:00    17.719999    17.719999    17.719999    17.719999
1928-01-05 00:00:00-05:00    17.549999    17.549999    17.549999    17.549999
1928-01-06 00:00:00-05:00    17.660000    17.660000    17.660000    17.660000
…                                  …            …            …            …
2023-10-20 00:00:00-04:00  4273.850098  4276.560059  4223.029785  4224.160156
2023-10-23 00:00:00-04:00  4210.399902  4255.839844  4189.220215  4217.040039
2023-10-24 00:00:00-04:00  4235.790039  4259.379883  4219.430176  4247.680176
2023-10-25 00:00:00-04:00  4232.419922  4232.419922  4181.419922  4186.770020
2023-10-26 00:00:00-04:00  4175.990234  4183.600098  4133.589844  4134.509766

                               Volume  Dividends  Stock Splits
Date
1927-12-30 00:00:00-05:00           0        0.0           0.0
1928-01-03 00:00:00-05:00           0        0.0           0.0
```

```
1928-01-04 00:00:00-05:00           0        0.0            0.0
1928-01-05 00:00:00-05:00           0        0.0            0.0
1928-01-06 00:00:00-05:00           0        0.0            0.0
…                                 …        …           …
2023-10-20 00:00:00-04:00  4004030000        0.0            0.0
2023-10-23 00:00:00-04:00  3776100000        0.0            0.0
2023-10-24 00:00:00-04:00  3821820000        0.0            0.0
2023-10-25 00:00:00-04:00  3869370000        0.0            0.0
2023-10-26 00:00:00-04:00  1356383000        0.0            0.0

[24071 rows x 7 columns]
```

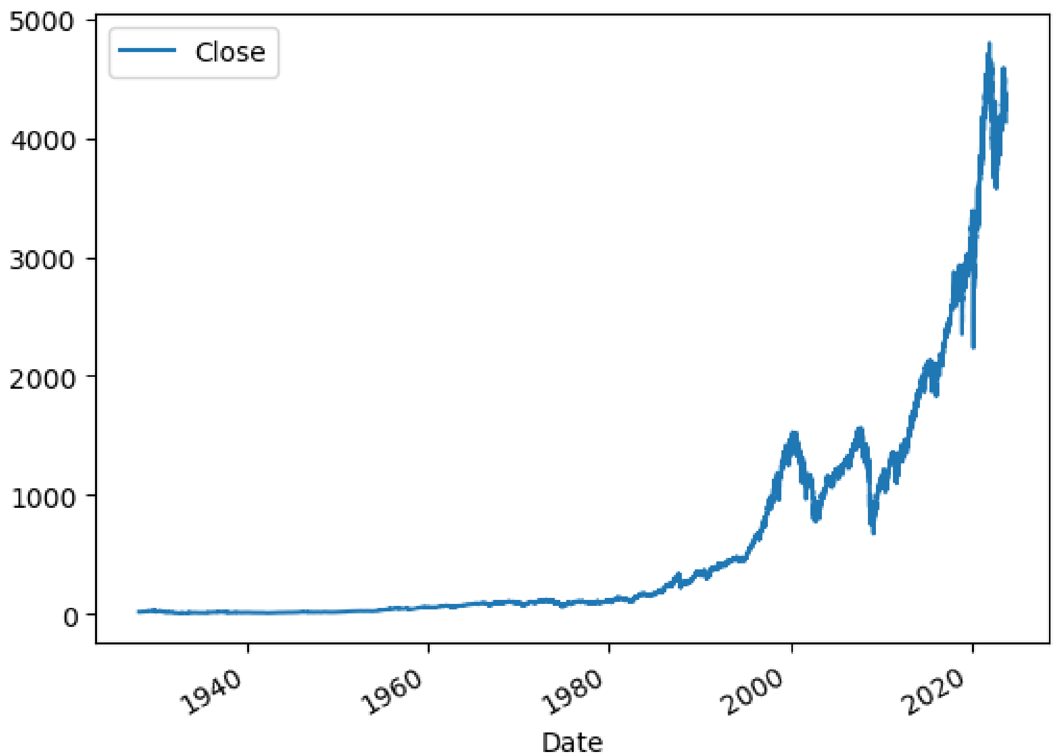
[9]:
```
sp500.index
```

[9]:
```
DatetimeIndex(['1927-12-30 00:00:00-05:00', '1928-01-03 00:00:00-05:00',
               '1928-01-04 00:00:00-05:00', '1928-01-05 00:00:00-05:00',
               '1928-01-06 00:00:00-05:00', '1928-01-09 00:00:00-05:00',
               '1928-01-10 00:00:00-05:00', '1928-01-11 00:00:00-05:00',
               '1928-01-12 00:00:00-05:00', '1928-01-13 00:00:00-05:00',
               …
               '2023-10-13 00:00:00-04:00', '2023-10-16 00:00:00-04:00',
               '2023-10-17 00:00:00-04:00', '2023-10-18 00:00:00-04:00',
               '2023-10-19 00:00:00-04:00', '2023-10-20 00:00:00-04:00',
               '2023-10-23 00:00:00-04:00', '2023-10-24 00:00:00-04:00',
               '2023-10-25 00:00:00-04:00', '2023-10-26 00:00:00-04:00'],
              dtype='datetime64[ns, America/New_York]', name='Date',
length=24071, freq=None)
```

Cleaning and Visualizing our stock market data

[10]:
```
sp500.plot.line(y='Close',use_index=True)
```

[10]:
```
<Axes: xlabel='Date'>
```



[12]:
```
del sp500['Dividends']
```

[13]:
```
del sp500['Stock Splits']
```

[14]:
```
sp500.head()
```

[14]:
```
                                Open       High        Low      Close  Volume
Date
1927-12-30 00:00:00-05:00  17.660000  17.660000  17.660000  17.660000       0
1928-01-03 00:00:00-05:00  17.760000  17.760000  17.760000  17.760000       0
1928-01-04 00:00:00-05:00  17.719999  17.719999  17.719999  17.719999       0
1928-01-05 00:00:00-05:00  17.549999  17.549999  17.549999  17.549999       0
1928-01-06 00:00:00-05:00  17.660000  17.660000  17.660000  17.660000       0
```

[15]:
```
sp500.tail()
```

[15]:
```
                                  Open         High          Low        Close  \
Date
2023-10-20 00:00:00-04:00  4273.850098  4276.560059  4223.029785  4224.160156
2023-10-23 00:00:00-04:00  4210.399902  4255.839844  4189.220215  4217.040039
2023-10-24 00:00:00-04:00  4235.790039  4259.379883  4219.430176  4247.680176
2023-10-25 00:00:00-04:00  4232.419922  4232.419922  4181.419922  4186.770020
```

```
2023-10-26 00:00:00-04:00  4175.990234  4183.600098  4133.589844  4134.509766

                               Volume
Date
2023-10-20 00:00:00-04:00  4004030000
2023-10-23 00:00:00-04:00  3776100000
2023-10-24 00:00:00-04:00  3821820000
2023-10-25 00:00:00-04:00  3869370000
2023-10-26 00:00:00-04:00  1356383000
```

It is accurate based on the directionality

[17]:
```
sp500['Tomorrow']=sp500['Close'].shift(-1)
```

[18]:
```
sp500
```

[18]:
```
                                  Open         High          Low        Close  \
Date
1927-12-30 00:00:00-05:00    17.660000    17.660000    17.660000    17.660000
1928-01-03 00:00:00-05:00    17.760000    17.760000    17.760000    17.760000
1928-01-04 00:00:00-05:00    17.719999    17.719999    17.719999    17.719999
1928-01-05 00:00:00-05:00    17.549999    17.549999    17.549999    17.549999
1928-01-06 00:00:00-05:00    17.660000    17.660000    17.660000    17.660000
…                                  …          …          …          …
2023-10-20 00:00:00-04:00  4273.850098  4276.560059  4223.029785  4224.160156
2023-10-23 00:00:00-04:00  4210.399902  4255.839844  4189.220215  4217.040039
2023-10-24 00:00:00-04:00  4235.790039  4259.379883  4219.430176  4247.680176
2023-10-25 00:00:00-04:00  4232.419922  4232.419922  4181.419922  4186.770020
2023-10-26 00:00:00-04:00  4175.990234  4183.600098  4133.589844  4134.509766

                               Volume     Tomorrow
Date
1927-12-30 00:00:00-05:00           0    17.760000
1928-01-03 00:00:00-05:00           0    17.719999
1928-01-04 00:00:00-05:00           0    17.549999
1928-01-05 00:00:00-05:00           0    17.660000
1928-01-06 00:00:00-05:00           0    17.500000
…                                …          …
2023-10-20 00:00:00-04:00  4004030000  4217.040039
2023-10-23 00:00:00-04:00  3776100000  4247.680176
2023-10-24 00:00:00-04:00  3821820000  4186.770020
2023-10-25 00:00:00-04:00  3869370000  4134.509766
2023-10-26 00:00:00-04:00  1356383000          NaN

[24071 rows x 6 columns]
```

[21]:
```
sp500['Target']=(sp500['Tomorrow']>sp500['Close']).astype(int)
```

[22]:
```
sp500
```

[22]:
```
                               Open         High          Low        Close  \
Date
1927-12-30 00:00:00-05:00    17.660000    17.660000    17.660000    17.660000
1928-01-03 00:00:00-05:00    17.760000    17.760000    17.760000    17.760000
1928-01-04 00:00:00-05:00    17.719999    17.719999    17.719999    17.719999
1928-01-05 00:00:00-05:00    17.549999    17.549999    17.549999    17.549999
1928-01-06 00:00:00-05:00    17.660000    17.660000    17.660000    17.660000
…                                  …            …            …            …
2023-10-20 00:00:00-04:00  4273.850098  4276.560059  4223.029785  4224.160156
2023-10-23 00:00:00-04:00  4210.399902  4255.839844  4189.220215  4217.040039
2023-10-24 00:00:00-04:00  4235.790039  4259.379883  4219.430176  4247.680176
2023-10-25 00:00:00-04:00  4232.419922  4232.419922  4181.419922  4186.770020
2023-10-26 00:00:00-04:00  4175.990234  4183.600098  4133.589844  4134.509766

                               Volume     Tomorrow  Target
Date
1927-12-30 00:00:00-05:00           0    17.760000       1
1928-01-03 00:00:00-05:00           0    17.719999       0
1928-01-04 00:00:00-05:00           0    17.549999       0
1928-01-05 00:00:00-05:00           0    17.660000       1
1928-01-06 00:00:00-05:00           0    17.500000       0
…                                   …          …       …
2023-10-20 00:00:00-04:00  4004030000  4217.040039       0
2023-10-23 00:00:00-04:00  3776100000  4247.680176       1
2023-10-24 00:00:00-04:00  3821820000  4186.770020       0
2023-10-25 00:00:00-04:00  3869370000  4134.509766       0
2023-10-26 00:00:00-04:00  1356383000          NaN       0

[24071 rows x 7 columns]
```

[35]:
```
sp500 = sp500.loc["1990-01-01":].copy()
#we use copy() we get the pandas copy() warning so .copy() helos to avoid that
```

[36]:
```
sp500
```

[36]:
```
                               Open         High          Low        Close  \
Date
1990-01-02 00:00:00-05:00   353.399994   359.690002   351.980011   359.690002
1990-01-03 00:00:00-05:00   359.690002   360.589996   357.890015   358.760010
1990-01-04 00:00:00-05:00   358.760010   358.760010   352.890015   355.670013
1990-01-05 00:00:00-05:00   355.670013   355.670013   351.350006   352.200012
1990-01-08 00:00:00-05:00   352.200012   354.239990   350.540009   353.790009
…                                  …            …            …            …
2023-10-20 00:00:00-04:00  4273.850098  4276.560059  4223.029785  4224.160156
2023-10-23 00:00:00-04:00  4210.399902  4255.839844  4189.220215  4217.040039
```

```
2023-10-24 00:00:00-04:00  4235.790039  4259.379883  4219.430176  4247.680176
2023-10-25 00:00:00-04:00  4232.419922  4232.419922  4181.419922  4186.770020
2023-10-26 00:00:00-04:00  4175.990234  4183.600098  4133.589844  4134.509766

                               Volume     Tomorrow  Target
Date
1990-01-02 00:00:00-05:00   162070000   358.760010       0
1990-01-03 00:00:00-05:00   192330000   355.670013       0
1990-01-04 00:00:00-05:00   177000000   352.200012       0
1990-01-05 00:00:00-05:00   158530000   353.790009       1
1990-01-08 00:00:00-05:00   140110000   349.619995       0
…                                 …          …     …
2023-10-20 00:00:00-04:00  4004030000  4217.040039       0
2023-10-23 00:00:00-04:00  3776100000  4247.680176       1
2023-10-24 00:00:00-04:00  3821820000  4186.770020       0
2023-10-25 00:00:00-04:00  3869370000  4134.509766       0
2023-10-26 00:00:00-04:00  1356383000          NaN       0

[8521 rows x 7 columns]
```

Training the inital ML model

[37]:
```
from sklearn.ensemble import RandomForestClassifier
```

[40]:
```
model =␣
↪RandomForestClassifier(n_estimators=100,min_samples_split=100,random_state=1)
```

[41]:
```
train=sp500.iloc[:-100]
test=sp500.iloc[-100:]
```

[42]:
```
predictor=['Close','Volume','Open','High','Low']
model.fit(train[predictor],train['Target'])
```

[42]:
```
RandomForestClassifier(min_samples_split=100, random_state=1)
```

[43]:
```
from sklearn.metrics import precision_score
```

[44]:
```
pred=model.predict(test[predictor])
```

[45]:
```
pred
```

[45]:
```
array([1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1,
       0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0,
       0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1,
       1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0])
```

[47]:
```
import pandas as pd
pred=pd.Series(pred,index=test.index)
```

[48]:
```
pred
```

[48]:
```
Date
2023-06-06 00:00:00-04:00    1
2023-06-07 00:00:00-04:00    1
2023-06-08 00:00:00-04:00    1
2023-06-09 00:00:00-04:00    1
2023-06-12 00:00:00-04:00    1
                            ..
2023-10-20 00:00:00-04:00    1
2023-10-23 00:00:00-04:00    1
2023-10-24 00:00:00-04:00    1
2023-10-25 00:00:00-04:00    1
2023-10-26 00:00:00-04:00    0
Length: 100, dtype: int32
```

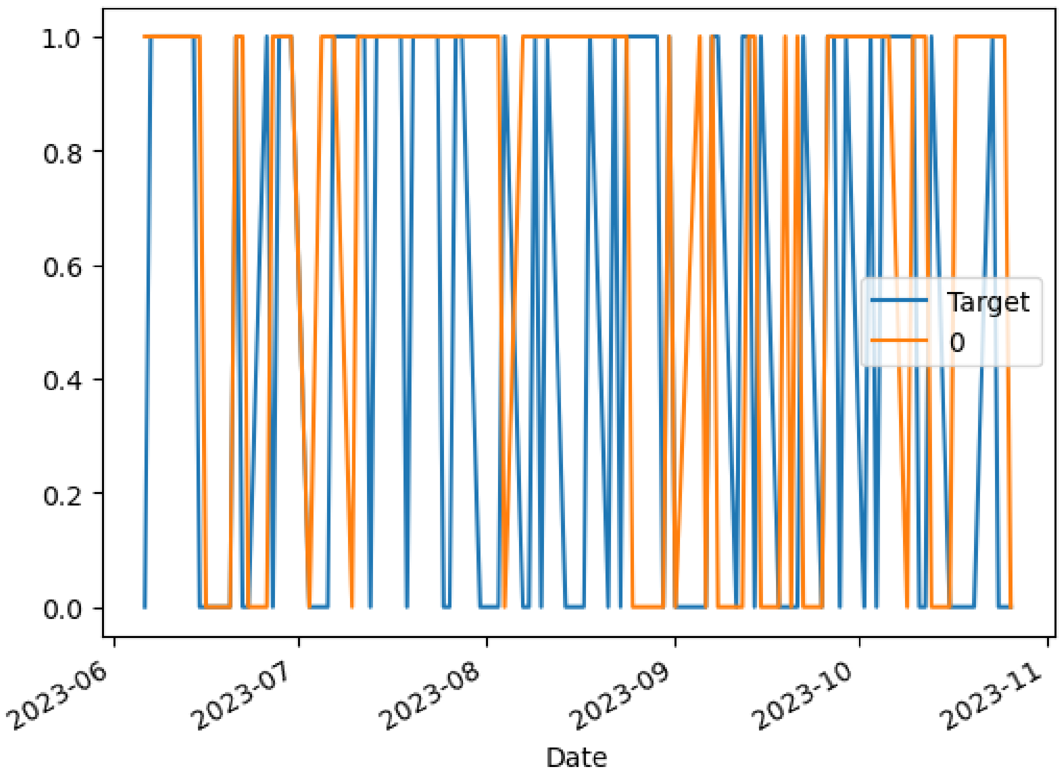
[50]:
```
precision_score(test['Target'],pred)
```

[50]:
```
0.49333333333333335
```

[52]:
```
combined =pd.concat([test["Target"],pred],axis=1)
```

[53]:
```
combined.plot()
```

[53]:
```
<Axes: xlabel='Date'>
```



Building Bcktesting System

[54]:
```
def predict(train,test,predictor,model):
    model.fit(train[predictor],train['Target'])
    pred=model.predict(test[predictor])
    pred=pd.Series(pred,index=test.index,name='Predictions')
    combined =pd.concat([test["Target"],pred],axis=1)
    return combined
```

[58]:
```
def backtest(data,model,predictor,start=2500,step=250):
# certain amount of data to train model every trading has 250 days
    all_prediction=[]


    for i in range(start,data.shape[0],step):
        train=data.iloc[0:i].copy()
        test = data.iloc[i:(i+step)].copy()
        prediction=predict(train,test,predictor,model)
        all_prediction.append(prediction)
    return pd.concat(all_prediction)
```

[59]:
```
prediction=backtest(sp500,model,predictor)
```

[72]:
```python
prediction["Predictions"].value_counts()
```

[72]:
```
0    3441
1    2580
Name: Predictions, dtype: int64
```

[71]:
```python
precision_score(prediction["Target"],prediction["Predictions"])
```

[71]:
```
0.5275193798449612
```

[76]:
```python
prediction["Target"].value_counts()/prediction.shape[0]
```

[76]:
```
1    0.533134
0    0.466866
Name: Target, dtype: float64
```

Adding more predicitors

[77]:
```python
horizons=[2,5,60,250,1000]
# last 2 days
# last Trading week
# last 3 moths of trading
# last year
# past years
new_predictors=[]

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

    ratio_column=f"Close_Ratio {horizon}"
    sp500[ratio_column]=sp500['Close']/rolling_avg['Close']

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

    new_predictors +=[ratio_column,trend_column]
```

[78]:
```python
sp500
```

[78]:
```
                                  Open         High          Low        Close  \
Date
1990-01-02 00:00:00-05:00   353.399994   359.690002   351.980011   359.690002
1990-01-03 00:00:00-05:00   359.690002   360.589996   357.890015   358.760010
1990-01-04 00:00:00-05:00   358.760010   358.760010   352.890015   355.670013
1990-01-05 00:00:00-05:00   355.670013   355.670013   351.350006   352.200012
1990-01-08 00:00:00-05:00   352.200012   354.239990   350.540009   353.790009
…                                    …            …            …            …
2023-10-20 00:00:00-04:00  4273.850098  4276.560059  4223.029785  4224.160156
```

```
2023-10-23 00:00:00-04:00  4210.399902  4255.839844  4189.220215  4217.040039
2023-10-24 00:00:00-04:00  4235.790039  4259.379883  4219.430176  4247.680176
2023-10-25 00:00:00-04:00  4232.419922  4232.419922  4181.419922  4186.770020
2023-10-26 00:00:00-04:00  4175.990234  4183.600098  4133.589844  4134.509766

                               Volume     Tomorrow  Target  Close_Ratio 2  \
Date
1990-01-02 00:00:00-05:00   162070000   358.760010       0            NaN
1990-01-03 00:00:00-05:00   192330000   355.670013       0       0.998706
1990-01-04 00:00:00-05:00   177000000   352.200012       0       0.995675
1990-01-05 00:00:00-05:00   158530000   353.790009       1       0.995098
1990-01-08 00:00:00-05:00   140110000   349.619995       0       1.002252
…                                   …            …       …              …
2023-10-20 00:00:00-04:00  4004030000  4217.040039       0       0.993668
2023-10-23 00:00:00-04:00  3776100000  4247.680176       1       0.999157
2023-10-24 00:00:00-04:00  3821820000  4186.770020       0       1.003620
2023-10-25 00:00:00-04:00  3869370000  4134.509766       0       0.992778
2023-10-26 00:00:00-04:00  1356383000          NaN       0       0.993720

                           Trend_2  Close_Ratio 5  Trend_5  Close_Ratio 60  \
Date
1990-01-02 00:00:00-05:00      NaN            NaN      NaN             NaN
1990-01-03 00:00:00-05:00      NaN            NaN      NaN             NaN
1990-01-04 00:00:00-05:00      0.0            NaN      NaN             NaN
1990-01-05 00:00:00-05:00      0.0            NaN      NaN             NaN
1990-01-08 00:00:00-05:00      1.0       0.993731      NaN             NaN
…                              …              …        …               …
2023-10-20 00:00:00-04:00      0.0       0.979466      1.0        0.958104
2023-10-23 00:00:00-04:00      0.0       0.984968      0.0        0.957811
2023-10-24 00:00:00-04:00      1.0       0.997976      1.0        0.966019
2023-10-25 00:00:00-04:00      1.0       0.989609      1.0        0.953576
2023-10-26 00:00:00-04:00      0.0       0.983931      1.0        0.943029

                           Trend_60  Close_Ratio 250  Trend_250  \
Date
1990-01-02 00:00:00-05:00       NaN              NaN        NaN
1990-01-03 00:00:00-05:00       NaN              NaN        NaN
1990-01-04 00:00:00-05:00       NaN              NaN        NaN
1990-01-05 00:00:00-05:00       NaN              NaN        NaN
1990-01-08 00:00:00-05:00       NaN              NaN        NaN
…                               …              …          …
2023-10-20 00:00:00-04:00      27.0         1.013671      124.0
2023-10-23 00:00:00-04:00      26.0         1.011555      123.0
2023-10-24 00:00:00-04:00      26.0         1.018525      123.0
2023-10-25 00:00:00-04:00      26.0         1.003576      123.0
2023-10-26 00:00:00-04:00      26.0         0.990739      123.0
```

```
                           Close_Ratio_1000  Trend_1000
Date
1990-01-02 00:00:00-05:00               NaN         NaN
1990-01-03 00:00:00-05:00               NaN         NaN
1990-01-04 00:00:00-05:00               NaN         NaN
1990-01-05 00:00:00-05:00               NaN         NaN
1990-01-08 00:00:00-05:00               NaN         NaN
…                                     …           …
2023-10-20 00:00:00-04:00          1.081485       528.0
2023-10-23 00:00:00-04:00          1.079336       528.0
2023-10-24 00:00:00-04:00          1.086850       528.0
2023-10-25 00:00:00-04:00          1.070961       527.0
2023-10-26 00:00:00-04:00          1.057307       527.0

[8521 rows x 17 columns]
```

[79]:
```
sp500=sp500.dropna()
```

[80]:
```
sp500
```

[80]:
```
                                  Open         High          Low        Close  \
Date
1993-12-14 00:00:00-05:00   465.730011   466.119995   462.459991   463.059998
1993-12-15 00:00:00-05:00   463.059998   463.690002   461.839996   461.839996
1993-12-16 00:00:00-05:00   461.859985   463.980011   461.859985   463.339996
1993-12-17 00:00:00-05:00   463.339996   466.380005   463.339996   466.380005
1993-12-20 00:00:00-05:00   466.380005   466.899994   465.529999   465.850006
…                                  …           …           …           …
2023-10-19 00:00:00-04:00  4321.359863  4339.540039  4269.689941  4278.000000
2023-10-20 00:00:00-04:00  4273.850098  4276.560059  4223.029785  4224.160156
2023-10-23 00:00:00-04:00  4210.399902  4255.839844  4189.220215  4217.040039
2023-10-24 00:00:00-04:00  4235.790039  4259.379883  4219.430176  4247.680176
2023-10-25 00:00:00-04:00  4232.419922  4232.419922  4181.419922  4186.770020

                               Volume     Tomorrow  Target  Close_Ratio_2  \
Date
1993-12-14 00:00:00-05:00   275050000   461.839996       0       0.997157
1993-12-15 00:00:00-05:00   331770000   463.339996       1       0.998681
1993-12-16 00:00:00-05:00   284620000   466.380005       1       1.001621
1993-12-17 00:00:00-05:00   363750000   465.850006       0       1.003270
1993-12-20 00:00:00-05:00   255900000   465.299988       0       0.999431
…                                 …          …      …             …
2023-10-19 00:00:00-04:00  3969730000  4224.160156       0       0.995741
2023-10-20 00:00:00-04:00  4004030000  4217.040039       0       0.993668
2023-10-23 00:00:00-04:00  3776100000  4247.680176       1       0.999157
2023-10-24 00:00:00-04:00  3821820000  4186.770020       0       1.003620
2023-10-25 00:00:00-04:00  3869370000  4134.509766       0       0.992778
```

```
                           Trend_2  Close_Ratio 5  Trend_5  Close_Ratio 60  \
Date
1993-12-14 00:00:00-05:00      1.0       0.996617      1.0        1.000283
1993-12-15 00:00:00-05:00      0.0       0.995899      1.0        0.997329
1993-12-16 00:00:00-05:00      1.0       0.999495      2.0        1.000311
1993-12-17 00:00:00-05:00      2.0       1.004991      3.0        1.006561
1993-12-20 00:00:00-05:00      1.0       1.003784      2.0        1.005120
…                                …              …      …             …
2023-10-19 00:00:00-04:00      0.0       0.987206      1.0        0.969168
2023-10-20 00:00:00-04:00      0.0       0.979466      1.0        0.958104
2023-10-23 00:00:00-04:00      0.0       0.984968      0.0        0.957811
2023-10-24 00:00:00-04:00      1.0       0.997976      1.0        0.966019
2023-10-25 00:00:00-04:00      1.0       0.989609      1.0        0.953576

                           Trend_60  Close_Ratio 250  Trend_250  \
Date
1993-12-14 00:00:00-05:00      32.0         1.028047      127.0
1993-12-15 00:00:00-05:00      32.0         1.025151      126.0
1993-12-16 00:00:00-05:00      32.0         1.028274      127.0
1993-12-17 00:00:00-05:00      32.0         1.034781      128.0
1993-12-20 00:00:00-05:00      32.0         1.033359      128.0
…                                …               …        …
2023-10-19 00:00:00-04:00      27.0         1.027055      125.0
2023-10-20 00:00:00-04:00      27.0         1.013671      124.0
2023-10-23 00:00:00-04:00      26.0         1.011555      123.0
2023-10-24 00:00:00-04:00      26.0         1.018525      123.0
2023-10-25 00:00:00-04:00      26.0         1.003576      123.0

                           Close_Ratio 1000  Trend_1000
Date
1993-12-14 00:00:00-05:00          1.176082       512.0
1993-12-15 00:00:00-05:00          1.172676       512.0
1993-12-16 00:00:00-05:00          1.176163       513.0
1993-12-17 00:00:00-05:00          1.183537       514.0
1993-12-20 00:00:00-05:00          1.181856       513.0
…                                       …         …
2023-10-19 00:00:00-04:00          1.095600       529.0
2023-10-20 00:00:00-04:00          1.081485       528.0
2023-10-23 00:00:00-04:00          1.079336       528.0
2023-10-24 00:00:00-04:00          1.086850       528.0
2023-10-25 00:00:00-04:00          1.070961       527.0

[7520 rows x 17 columns]
```

[81]:
```
model =
 ↪RandomForestClassifier(n_estimators=200,min_samples_split=50,random_state=1)
```

[87]:
```
def predict(train,test,predictor,model):
    model.fit(train[predictor],train['Target'])
    pred=model.predict_proba(test[predictor])[:,1]
    pred[pred>.6]=1
    pred[pred< .6]=0
    pred=pd.Series(pred,index=test.index,name='Predictions')
    combined =pd.concat([test["Target"],pred],axis=1)
    return combined
```

[88]:
```
prediction=backtest(sp500,model,new_predictors)
```

[89]:
```
prediction["Predictions"].value_counts()
```

[89]:
```
0.0    4196
1.0     824
Name: Predictions, dtype: int64
```

[91]:
```
precision_score(prediction['Target'],prediction['Predictions'])
```

[91]:
```
0.5703883495145631
```

[ ]:
```
```

1 Alphavantage API

API for real time stock data

set up environment

[92]:
```
!pip install alpha_vantage
```

```
Collecting alpha_vantage
  Downloading alpha_vantage-2.3.1-py3-none-any.whl (31 kB)
Requirement already satisfied: aiohttp in c:\users\morea\anaconda3\lib\site-
packages (from alpha_vantage) (3.8.3)
Requirement already satisfied: requests in c:\users\morea\anaconda3\lib\site-
packages (from alpha_vantage) (2.31.0)
Requirement already satisfied: attrs>=17.3.0 in
c:\users\morea\anaconda3\lib\site-packages (from aiohttp->alpha_vantage)
(22.1.0)
Requirement already satisfied: charset-normalizer<3.0,>=2.0 in
c:\users\morea\anaconda3\lib\site-packages (from aiohttp->alpha_vantage) (2.0.4)
Requirement already satisfied: multidict<7.0,>=4.5 in
c:\users\morea\anaconda3\lib\site-packages (from aiohttp->alpha_vantage) (6.0.2)
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in
c:\users\morea\anaconda3\lib\site-packages (from aiohttp->alpha_vantage) (4.0.2)
Requirement already satisfied: yarl<2.0,>=1.0 in
c:\users\morea\anaconda3\lib\site-packages (from aiohttp->alpha_vantage) (1.8.1)
```

```
Requirement already satisfied: frozenlist>=1.1.1 in
c:\users\morea\anaconda3\lib\site-packages (from aiohttp->alpha_vantage) (1.3.3)
Requirement already satisfied: aiosignal>=1.1.2 in
c:\users\morea\anaconda3\lib\site-packages (from aiohttp->alpha_vantage) (1.2.0)
Requirement already satisfied: idna<4,>=2.5 in
c:\users\morea\anaconda3\lib\site-packages (from requests->alpha_vantage) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in
c:\users\morea\anaconda3\lib\site-packages (from requests->alpha_vantage)
(1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in
c:\users\morea\anaconda3\lib\site-packages (from requests->alpha_vantage)
(2023.7.22)
Installing collected packages: alpha_vantage
Successfully installed alpha_vantage-2.3.1
```

[115]:
```
import pandas as pd
import matplotlib.pyplot as plt
from alpha_vantage.timeseries import TimeSeries
```

[116]:
```
#Store API key
key = open('Alphavantagekey.txt').read()
```

Creating TimeSeries object and make API cell

[117]:
```
ts=TimeSeries(key,output_format='pandas')
data,meta =ts.get_intraday('TSLA',interval='1min',outputsize='full')
```

Examin data

[118]:
```
meta
```

[118]:
```
{'1. Information': 'Intraday (1min) open, high, low, close prices and volume',
 '2. Symbol': 'TSLA',
 '3. Last Refreshed': '2023-10-25 19:59:00',
 '4. Interval': '1min',
 '5. Output Size': 'Full size',
 '6. Time Zone': 'US/Eastern'}
```

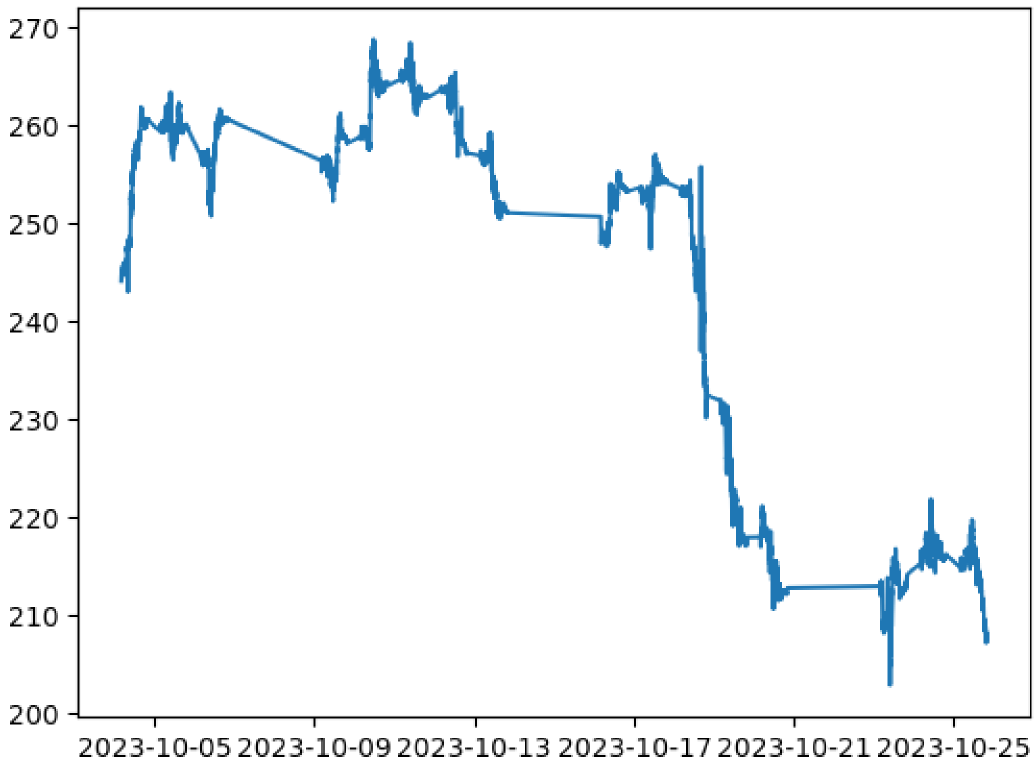
[119]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 15360 entries, 2023-10-25 19:59:00 to 2023-10-04 04:00:00
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   1. open    15360 non-null  float64
 1   2. high    15360 non-null  float64
 2   3. low     15360 non-null  float64
 3   4. close   15360 non-null  float64
```

```
 4   5. volume  15360 non-null  float64
dtypes: float64(5)
memory usage: 720.0 KB
```

[120]:
```python
plt.plot(data['4. close'])
```

[120]:
```
[<matplotlib.lines.Line2D at 0x27fef14eb50>]
```



[121]:
```python
data.head()
```

[121]:

date	1. open	2. high	3. low	4. close	5. volume
2023-10-25 19:59:00	207.80	207.85	207.50	207.73	13407.0
2023-10-25 19:58:00	207.75	207.85	207.55	207.77	7264.0
2023-10-25 19:57:00	207.75	207.84	207.65	207.78	6321.0
2023-10-25 19:56:00	207.70	207.88	207.65	207.76	5173.0
2023-10-25 19:55:00	207.68	207.75	207.56	207.70	3395.0

[122]:
```python
columns=['open','high','low','close','volume']
data.columns=columns
```

[123]:
```
data['TradeDate']=data.index.date
data['time']=data.index.time
```

[129]:
```
data.loc['2022-12-31']
```

[129]:
```
Empty DataFrame
Columns: [open, high, low, close, volume, TradeDate, time]
Index: []
```

[126]:
```
market=data.between_time('9:30:00','16:00:00').copy()
```

[127]:
```
market.sort_index(inplace=True)
market.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 6256 entries, 2023-10-04 09:30:00 to 2023-10-25 16:00:00
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   open       6256 non-null   float64
 1   high       6256 non-null   float64
 2   low        6256 non-null   float64
 3   close      6256 non-null   float64
 4   volume     6256 non-null   float64
 5   TradeDate  6256 non-null   object
 6   time       6256 non-null   object
dtypes: float64(5), object(2)
memory usage: 391.0+ KB
```

[128]:
```
market.groupby('TradeDate').agg({'low':min,'high':max})
```

[128]:
```
                low     high
TradeDate
2023-10-04  247.600  261.860
2023-10-05  256.250  263.600
2023-10-06  250.650  261.650
2023-10-09  252.050  261.360
2023-10-10  257.650  268.940
2023-10-11  260.900  268.600
2023-10-12  256.631  265.410
2023-10-13  250.220  259.600
2023-10-16  248.480  255.400
2023-10-17  247.080  257.183
2023-10-18  241.800  254.630
2023-10-19  216.780  230.610
2023-10-20  210.420  218.861
2023-10-23  202.510  216.980
2023-10-24  214.110  222.050
```

```
2023-10-25  212.200  220.100
```

[134]:
```python
market.loc[market.groupby('TradeDate')['low'].idxmin()]
```

[134]:
```
                        open     high      low    close     volume  \
date
2023-10-04 09:30:00  248.140  248.680  247.600  248.250   823866.0
2023-10-05 11:10:00  256.460  256.990  256.250  256.915   466539.0
2023-10-06 09:56:00  251.340  251.600  250.650  250.690   680249.0
2023-10-09 11:10:00  252.410  252.430  252.050  252.201   290406.0
2023-10-10 09:30:00  257.750  258.980  257.650  258.470   849728.0
2023-10-11 13:40:00  260.989  261.184  260.900  261.169   216959.0
2023-10-12 14:13:00  256.800  256.870  256.631  256.786   276725.0
2023-10-13 15:06:00  251.020  251.050  250.220  250.650   421652.0
2023-10-16 09:30:00  250.050  250.680  248.480  249.649  1292122.0
2023-10-17 09:54:00  247.371  247.690  247.080  247.601   379813.0
2023-10-18 16:00:00  242.550  243.500  241.800  242.680  4938246.0
2023-10-19 14:56:00  217.002  217.080  216.780  217.080   506039.0
2023-10-20 11:35:00  210.680  210.930  210.420  210.880   657010.0
2023-10-23 09:44:00  203.540  203.660  202.510  202.847  1182987.0
2023-10-24 12:43:00  214.290  214.350  214.110  214.348   292706.0
2023-10-25 15:52:00  212.489  212.540  212.200  212.440   332149.0

                      TradeDate      time
date
2023-10-04 09:30:00  2023-10-04  09:30:00
2023-10-05 11:10:00  2023-10-05  11:10:00
2023-10-06 09:56:00  2023-10-06  09:56:00
2023-10-09 11:10:00  2023-10-09  11:10:00
2023-10-10 09:30:00  2023-10-10  09:30:00
2023-10-11 13:40:00  2023-10-11  13:40:00
2023-10-12 14:13:00  2023-10-12  14:13:00
2023-10-13 15:06:00  2023-10-13  15:06:00
2023-10-16 09:30:00  2023-10-16  09:30:00
2023-10-17 09:54:00  2023-10-17  09:54:00
2023-10-18 16:00:00  2023-10-18  16:00:00
2023-10-19 14:56:00  2023-10-19  14:56:00
2023-10-20 11:35:00  2023-10-20  11:35:00
2023-10-23 09:44:00  2023-10-23  09:44:00
2023-10-24 12:43:00  2023-10-24  12:43:00
2023-10-25 15:52:00  2023-10-25  15:52:00
```

[135]:
```python
market.loc[market.groupby('TradeDate')['high'].idxmax()]
```

[135]:
```
                        open     high      low    close     volume  \
date
2023-10-04 15:47:00  261.600  261.860  261.530  261.800   441969.0
```

```
2023-10-05 09:36:00  263.340  263.600  262.860  263.113   967195.0
2023-10-06 15:09:00  261.261  261.650  261.250  261.600   426775.0
2023-10-09 15:33:00  261.190  261.360  261.080  261.090   415725.0
2023-10-10 11:32:00  268.655  268.940  268.545  268.721   396547.0
2023-10-11 09:36:00  268.390  268.600  268.100  268.160   693789.0
2023-10-12 09:33:00  264.650  265.410  264.200  264.279   849994.0
2023-10-13 09:30:00  258.900  259.600  258.500  259.163   832711.0
2023-10-16 14:20:00  255.180  255.400  255.161  255.239   248016.0
2023-10-17 12:42:00  256.910  257.183  256.860  256.990   273899.0
2023-10-18 09:35:00  254.260  254.630  253.920  254.020   453776.0
2023-10-19 09:33:00  230.080  230.610  229.090  229.180  1335514.0
2023-10-20 10:01:00  218.280  218.861  218.220  218.245   713578.0
2023-10-23 13:00:00  216.730  216.980  216.440  216.590   324484.0
2023-10-24 10:10:00  221.775  222.050  221.140  221.240   808705.0
2023-10-25 11:15:00  219.640  220.100  219.415  219.510   791522.0

                      TradeDate      time
date
2023-10-04 15:47:00  2023-10-04  15:47:00
2023-10-05 09:36:00  2023-10-05  09:36:00
2023-10-06 15:09:00  2023-10-06  15:09:00
2023-10-09 15:33:00  2023-10-09  15:33:00
2023-10-10 11:32:00  2023-10-10  11:32:00
2023-10-11 09:36:00  2023-10-11  09:36:00
2023-10-12 09:33:00  2023-10-12  09:33:00
2023-10-13 09:30:00  2023-10-13  09:30:00
2023-10-16 14:20:00  2023-10-16  14:20:00
2023-10-17 12:42:00  2023-10-17  12:42:00
2023-10-18 09:35:00  2023-10-18  09:35:00
2023-10-19 09:33:00  2023-10-19  09:33:00
2023-10-20 10:01:00  2023-10-20  10:01:00
2023-10-23 13:00:00  2023-10-23  13:00:00
2023-10-24 10:10:00  2023-10-24  10:10:00
2023-10-25 11:15:00  2023-10-25  11:15:00
```

[ ]:

[ ]:

[ ]: