

Untitled38

May 3, 2024

```
[5]: import yfinance as yf
import os
import pandas as pd
```

```
[7]: btc_ticker = yf.Ticker("BTC-USD")
```

```
[171]: if os.path.exists("btc.csv"):
    btc = pd.read_csv("btc.csv", index_col=0)
else:
    btc = btc_ticker.history(period="max")
    btc.to_csv("btc.csv")
```

```
[175]: btc
```

```
[175]:
```

	Open	High	Low \
Date			
2014-09-17 00:00:00+00:00	465.864014	468.174011	452.421997
2014-09-18 00:00:00+00:00	456.859985	456.859985	413.104004
2014-09-19 00:00:00+00:00	424.102997	427.834991	384.532013
2014-09-20 00:00:00+00:00	394.673004	423.295990	389.882996
2014-09-21 00:00:00+00:00	408.084991	412.425995	393.181000
...
2024-04-28 00:00:00+00:00	63423.515625	64321.484375	62793.597656
2024-04-29 00:00:00+00:00	63106.363281	64174.878906	61795.457031
2024-04-30 00:00:00+00:00	63839.417969	64703.332031	59120.066406
2024-05-01 00:00:00+00:00	60609.496094	60780.500000	56555.292969
2024-05-02 00:00:00+00:00	58267.679688	58498.878906	56985.296875

	Close	Volume	Dividends	Stock Splits
Date				
2014-09-17 00:00:00+00:00	457.334015	21056800	0.0	0.0
2014-09-18 00:00:00+00:00	424.440002	34483200	0.0	0.0
2014-09-19 00:00:00+00:00	394.795990	37919700	0.0	0.0
2014-09-20 00:00:00+00:00	408.903992	36863600	0.0	0.0
2014-09-21 00:00:00+00:00	398.821014	26580100	0.0	0.0
...
2024-04-28 00:00:00+00:00	63113.230469	17334827993	0.0	0.0

2024-04-29 00:00:00+00:00	63841.121094	26635912073	0.0	0.0
2024-04-30 00:00:00+00:00	60636.855469	37840840057	0.0	0.0
2024-05-01 00:00:00+00:00	58254.011719	48439780271	0.0	0.0
2024-05-02 00:00:00+00:00	57686.480469	45496893440	0.0	0.0

[3516 rows x 7 columns]

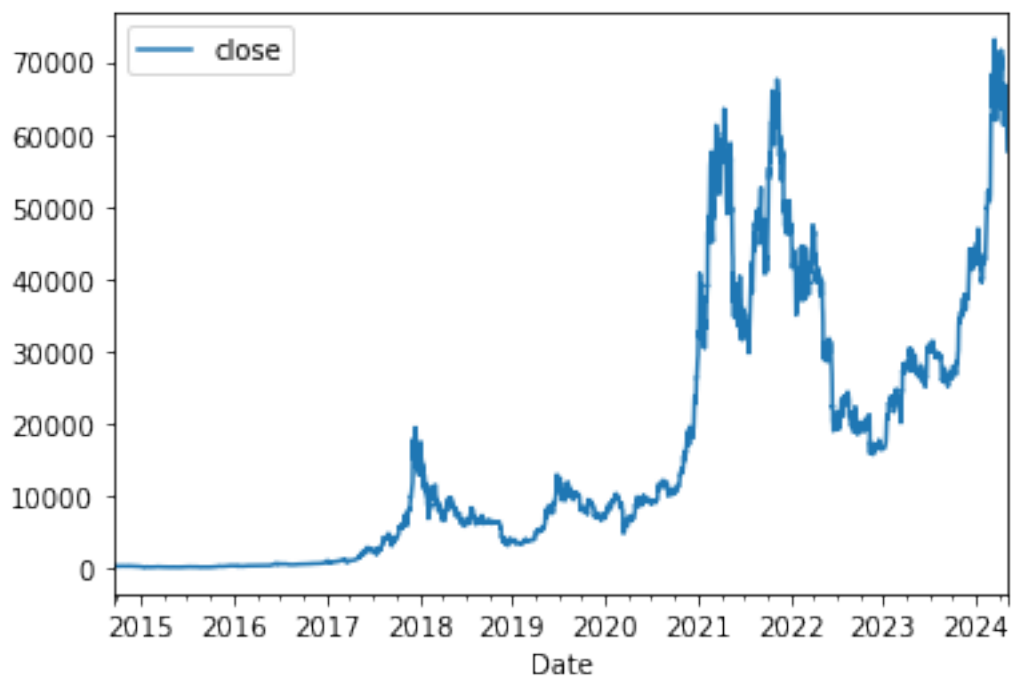
```
[177]: btc.index = pd.to_datetime(btc.index)
```

```
[179]: del btc["Dividends"]
del btc["Stock Splits"]
```

```
[218]: btc.columns = [c.lower() for c in btc.columns]
```

```
[220]: btc.plot.line(y="close", use_index=True)
```

```
[220]: <AxesSubplot:xlabel='Date'>
```



```
[183]: wiki = pd.read_csv("wikipedia_edits.csv", index_col=0, parse_dates=True)
```

```
[185]: wiki
```

```
[185]:
```

	edit_count	editor_count	ratio	difference	sentiment	\
2009-04-06	0.133333	0.066667	4.408755	4.633333	-0.032780	

2009-04-07	0.000000	0.000000	0.000000	0.000000	0.000000
2009-04-08	0.000000	0.000000	0.000000	0.000000	0.000000
2009-04-09	0.000000	0.000000	0.000000	0.000000	0.000000
2009-04-10	0.000000	0.000000	0.000000	0.000000	0.000000
...
2022-09-03	0.266667	0.200000	0.000092	20.588889	-0.092629
2022-09-04	0.266667	0.200000	0.000092	20.588889	-0.092629
2022-09-05	0.266667	0.200000	0.000092	20.588889	-0.092629
2022-09-06	0.266667	0.200000	0.000092	20.588889	-0.092629
2022-09-07	0.266667	0.200000	0.000092	20.588889	-0.092629

	neg_sentiment	total_difference
2009-04-06	0.033333	18.533333
2009-04-07	0.000000	0.000000
2009-04-08	0.000000	0.000000
2009-04-09	0.000000	0.000000
2009-04-10	0.000000	0.000000
...
2022-09-03	0.111111	40.200000
2022-09-04	0.111111	40.200000
2022-09-05	0.111111	40.200000
2022-09-06	0.111111	40.200000
2022-09-07	0.111111	40.200000

[4903 rows x 7 columns]

```
[187]: btc.index = btc.index.tz_localize(None)
       bbtc = btc.merge(wiki, left_index=True, right_index=True, how='inner')
```

```
[189]: btc["tomorrow"] = btc["close"].shift(-1)
```

```
[191]: btc["target"] = (btc["tomorrow"] > btc["close"]).astype(int)
```

```
[193]: btc["target"].value_counts()
```

```
[193]: target
1      1862
0      1654
Name: count, dtype: int64
```

```
[195]: btc
```

```
[195]:
```

	open	high	low	close \
Date				
2014-09-17	465.864014	468.174011	452.421997	457.334015
2014-09-18	456.859985	456.859985	413.104004	424.440002
2014-09-19	424.102997	427.834991	384.532013	394.795990

2014-09-20	394.673004	423.295990	389.882996	408.903992
2014-09-21	408.084991	412.425995	393.181000	398.821014
...
2024-04-28	63423.515625	64321.484375	62793.597656	63113.230469
2024-04-29	63106.363281	64174.878906	61795.457031	63841.121094
2024-04-30	63839.417969	64703.332031	59120.066406	60636.855469
2024-05-01	60609.496094	60780.500000	56555.292969	58254.011719
2024-05-02	58267.679688	58498.878906	56985.296875	57686.480469

Date	volume	tomorrow	target
2014-09-17	21056800	424.440002	0
2014-09-18	34483200	394.795990	0
2014-09-19	37919700	408.903992	1
2014-09-20	36863600	398.821014	0
2014-09-21	26580100	402.152008	1
...
2024-04-28	17334827993	63841.121094	1
2024-04-29	26635912073	60636.855469	0
2024-04-30	37840840057	58254.011719	0
2024-05-01	48439780271	57686.480469	0
2024-05-02	45496893440	NaN	0

[3516 rows x 7 columns]

```
[197]: import pandas as pd
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(n_estimators=100, min_samples_split=50,
                               random_state=1)

train = btc.iloc[:-200]
test = btc.iloc[-200:]

predictors = ["close", "volume", "open", "high", "low"]
model.fit(train[predictors], train["target"])
```

```
[197]: RandomForestClassifier(min_samples_split=50, random_state=1)
```

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

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

```
[198]: 0.56
```

```
[201]: 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
```

```
[203]: def backtest(data, model, predictors, start=1095, step=150):
        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)
```

```
[205]: from xgboost import XGBClassifier

        model = XGBClassifier(random_state=1, learning_rate=.1, n_estimators=200)
        predictions = backtest(btc, model, predictors)
```

```
[206]: predictions["predictions"].value_counts()
```

```
[206]: predictions
1      1464
0       957
Name: count, dtype: int64
```

```
[207]: precision_score(predictions["target"], predictions["predictions"])
```

```
[207]: 0.5013661202185792
```

```
[211]: def compute_rolling(btc):
        horizons = [2,7,60,365]
        new_predictors = ["close", "sentiment", "neg_sentiment"]

        for horizon in horizons:
            rolling_averages = btc.rolling(horizon, min_periods=1).mean()

            ratio_column = f"close_ratio_{horizon}"
            btc[ratio_column] = btc["close"] / rolling_averages["close"]

            edit_column = f"edit_{horizon}"
            btc[edit_column] = rolling_averages["edit_count"]
```

```

        rolling = btc.rolling(horizon, closed='left', min_periods=1).mean()
        trend_column = f"trend_{horizon}"
        btc[trend_column] = rolling["target"]

        new_predictors+= [ratio_column, trend_column, edit_column]
    return btc, new_predictors

```

```
[ ]: btc, new_predictors = compute_rolling(btc.copy())
```

```
[ ]: predictions = backtest(btc, model, new_predictors)
```

```
[223]: precision_score(predictions["target"], predictions["predictions"])
```

```
[223]: 0.5013661202185792
```

```
[225]: predictions
```

```
[225]:
```

	target	predictions
Date		
2017-09-16	0	1
2017-09-17	1	1
2017-09-18	0	1
2017-09-19	0	1
2017-09-20	0	1
...
2024-04-28	1	1
2024-04-29	0	1
2024-04-30	0	1
2024-05-01	0	0
2024-05-02	0	0

```
[2421 rows x 2 columns]
```

```
[227]: last_row = predictions.iloc[-1]
        print(last_row)
```

```

target      0
predictions 0
Name: 2024-05-02 00:00:00, dtype: int64

```