

Sec 1 Lecture #2 - Adrian

January 17, 2024

1 1.) Pull in Data and Convert to Monthly

```
[35]: import yfinance as yf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
[36]: apple_data = yf.download('AAPL')
df = apple_data.resample("M").last()[["Adj Close"]]
```

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

2 2.) Create columns

- Current Stock Price, Difference in stock price, Whether it went up or down over the next month, option premium

```
[37]: #difference for the next period
df["Diff"] = df["Adj Close"].diff().shift(-1)

df["Target"] = np.sign(df['Diff'])

df["Premium"] = .08 * df['Adj Close']
```

```
[38]: df.head()
```

```
[38]:
```

	Adj Close	Diff	Target	Premium
Date				
1980-12-31	0.117887	-0.020296	-1.0	0.009431
1981-01-31	0.097592	-0.006045	-1.0	0.007807
1981-02-28	0.091546	-0.006909	-1.0	0.007324
1981-03-31	0.084637	0.013386	1.0	0.006771
1981-04-30	0.098023	0.016409	1.0	0.007842

3 3.) Pull in X data, normalize and build a LogReg on column 2

```
[39]: import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
```

```
[40]: X = pd.read_csv("Xdata.csv", index_col="Date", parse_dates=["Date"])
```

```
[41]: y = df.loc[:, "2023-09-30", "Target"].copy()

df = df.loc[:, "2023-09-30", :].copy()
```

```
[42]: logreg = LogisticRegression()

logreg.fit(X, y)

y_pred = logreg.predict(X)
```

```
[44]: df.head()
```

```
[44]:
```

	Adj Close	Diff	Target	Premium
Date				
1980-12-31	0.117887	-0.020296	-1.0	0.009431
1981-01-31	0.097592	-0.006045	-1.0	0.007807
1981-02-28	0.091546	-0.006909	-1.0	0.007324
1981-03-31	0.084637	0.013386	1.0	0.006771
1981-04-30	0.098023	0.016409	1.0	0.007842

4 4.) Add columns, prediction, and profits

```
[46]: df["Prediction"] = y_pred
```

```
[50]: df["Profits"] = 0

# True Positive
# Every time correct for upside (only times it went up)
# Product of target and prediction

df.loc[(df["Prediction"] == 1) & (df["Target"] == 1), "Profits"] = df["Premium"]

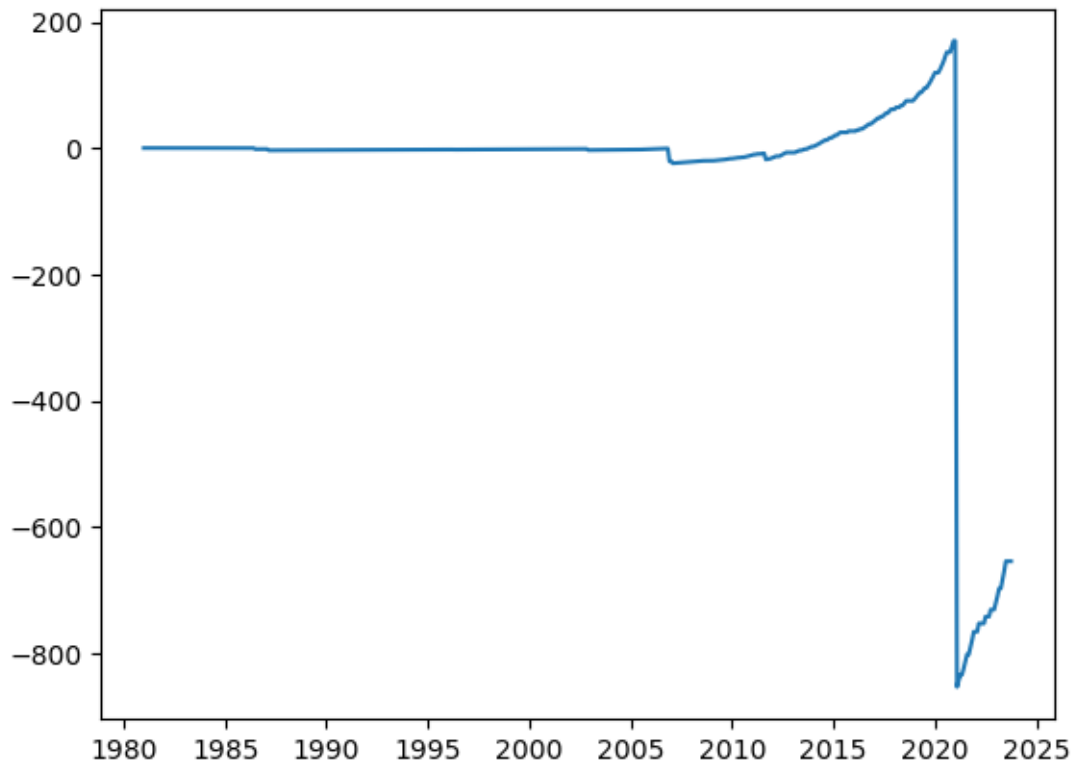
# False Positive

df.loc[(df["Prediction"] == 1) & (df["Target"] == -1), "Profits"] = 100 *
↳ df["Diff"] + df["Premium"] #Commission
```

5 5.) Plot profits over time

```
[49]: plt.plot(np.cumsum(df["Profits"]))
```

```
[49]: [<matplotlib.lines.Line2D at 0x29cbcc050>]
```



6 5.5) Your skills from the MQE to help Mr. Liu's ventures

As a current MQE student, the classes that I've took have been training me in how to interpret, analyze and deliver results. Mainly in statistical programming, the program has been teaching many useful skills such as econometrics, machine learning, as well as asset forecasting through the classes that I have/am taking so far.

Meanwhile, I have a couple of very recent experience which was completed last fall in my industry applied projects, where I worked with two different companies to deliver business solutions. One of them was with an investment management company, where I've succesfully built a tool to gather key business/industry metrics for every listed companies using large language model api to help equity researchers automate their data gathering for fundamental analysis. Another project, where I've worked with a multinational audio system company was to implement generative AI by creating a tool to improve e-commerce customer experience and data acquisition processes. (no further information available; subject to an NDA.)

7 6.) Create a loop that stores total profits over time

[]:

8 7.) What is the optimal threshold and plot the total profits for this model

[]:

<https://github.com/adrianonggo/UCLA>