

# Sec 1 Lecture 2-Logistic Regression

January 17, 2024

## 1 1.) Pull in Data and Convert ot Monthly

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

```
[60]: 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

```
[61]: # Create columns
df["Diff"] = df["Adj Close"].diff().shift(-1)
df["Up or Down"] = df["Diff"].apply(lambda x: "Up" if x > 0 else "Down")

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

# Option premium
df["Premium"] = 0.08 * df['Adj Close']

df
```

```
[61]:
```

	Adj Close	Diff	Up or Down	Target	Premium
Date					
1980-12-31	0.117887	-0.020296	Down	-1.0	0.009431
1981-01-31	0.097591	-0.006045	Down	-1.0	0.007807
1981-02-28	0.091546	-0.006909	Down	-1.0	0.007324
1981-03-31	0.084637	0.013386	Up	1.0	0.006771
1981-04-30	0.098023	0.016409	Up	1.0	0.007842
...	...	...	...	...	...
2023-09-30	170.984741	-0.439423	Down	-1.0	13.678779
2023-10-31	170.545319	19.404678	Up	1.0	13.643625

2023-11-30	189.949997	2.580002	Up	1.0	15.196000
2023-12-31	192.529999	-9.850006	Down	-1.0	15.402400
2024-01-31	182.679993	NaN	Down	NaN	14.614399

[518 rows x 5 columns]

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

```
[62]: 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
```

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

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

```
[65]: logreg= LogisticRegression()
logreg.fit(X,y)
y_pred = logreg.predict(X)
```

```
[66]: y_pred
```

```
[66]: array([-1., -1., -1.,  1.,  1., -1., -1., -1., -1.,  1., -1.,  1., -1.,
-1., -1., -1., -1., -1.,  1.,  1.,  1.,  1.,  1., -1.,  1.,  1.,
-1.,  1.,  1., -1., -1.,  1., -1., -1., -1.,  1.,  1.,  1., -1.,
  1., -1., -1.,  1.,  1., -1., -1., -1.,  1., -1., -1., -1., -1.,
-1.,  1., -1., -1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,
-1.,  1.,  1., -1.,  1.,  1.,  1.,  1.,  1.,  1.,  1., -1.,  1.,
  1.,  1.,  1., -1., -1.,  1., -1.,  1., -1.,  1.,  1.,  1., -1.,
-1.,  1., -1., -1.,  1., -1., -1., -1.,  1.,  1., -1., -1.,  1.,
-1.,  1., -1., -1., -1.,  1.,  1., -1.,  1.,  1., -1., -1., -1.,
  1.,  1.,  1.,  1.,  1.,  1., -1., -1., -1., -1.,  1., -1.,  1.,
-1.,  1.,  1.,  1., -1.,  1., -1., -1., -1., -1., -1.,  1.,  1.,
  1., -1., -1., -1., -1.,  1., -1., -1., -1., -1.,  1.,  1., -1.,
  1.,  1., -1., -1., -1., -1.,  1.,  1., -1.,  1., -1.,  1.,  1.,
-1., -1.,  1.,  1.,  1., -1., -1., -1., -1.,  1., -1., -1., -1.,
-1.,  1.,  1., -1.,  1.,  1., -1.,  1.,  1., -1., -1., -1.,  1.,
-1., -1., -1.,  1.,  1., -1., -1.,  1., -1.,  1.,  1.,  1., -1.,
-1.,  1.,  1., -1.,  1., -1., -1.,  1.,  1., -1.,  1.,  1., -1.,
  1.,  1.,  1., -1.,  1.,  1.,  1.,  1.,  1.,  1., -1., -1.,  1.,
-1.,  1., -1., -1., -1., -1.,  1., -1.,  1.,  1., -1.,  1., -1.,
-1., -1.,  1., -1.,  1.,  1., -1.,  1.,  1., -1., -1., -1., -1.,
-1.,  1., -1.,  1.,  1.,  1., -1.,  1., -1.,  1.,  1.,  1., -1.,
```

```

1., -1., 1., 1., 1., 1., -1., 1., 1., -1., 1., 1., 1.,
1., -1., 1., 1., -1., -1., 1., -1., 1., 1., 1., 1., 1.,
1., 1., -1., -1., 1., -1., -1., 1., -1., 1., 1., 1., 1.,
1., 1., 1., 1., 1., 1., -1., 1., 1., 1., -1., 1., -1.,
-1., 1., 1., 1., -1., -1., 1., -1., -1., -1., -1., 1., -1.,
1., 1., 1., -1., 1., 1., 1., 1., 1., 1., -1., 1., 1.,
1., -1., -1., 1., -1., 1., 1., 1., 1., 1., 1., -1., 1.,
-1., -1., 1., -1., 1., 1., -1., 1., 1., 1., 1., -1., -1.,
1., 1., 1., 1., -1., -1., -1., -1., -1., 1., 1., 1., -1.,
1., 1., -1., 1., 1., 1., -1., 1., 1., 1., 1., 1., 1.,
1., -1., 1., 1., -1., 1., 1., -1., 1., 1., -1., -1., -1.,
-1., 1., -1., -1., -1., -1., 1., -1., 1., -1., 1., 1., 1.,
1., -1., 1., 1., 1., 1., -1., 1., -1., 1., 1., -1., 1.,
1., -1., -1., 1., -1., -1., 1., -1., 1., 1., -1., -1., -1.,
-1., 1., 1., 1., 1., -1., 1., 1., -1., 1., 1., 1., 1.,
1., -1., -1., 1., 1., 1., 1., 1., -1., -1., 1., 1., -1.,
1., 1., 1., -1., 1., 1., 1., -1., 1., 1., 1., -1., -1.,
1., -1., -1., -1., 1., -1., -1., 1., -1., -1., 1., 1., 1.,
-1., 1., 1., 1., -1., -1., -1.]

```

#### 4 4.) Add columns, prediction and profits.

```
[67]: df["Predictions"] = y_pred
```

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

#True Positives
df.loc[(df["Predictions"] == 1) & (df["Target"] == 1), "Profits"] = df['Premium']

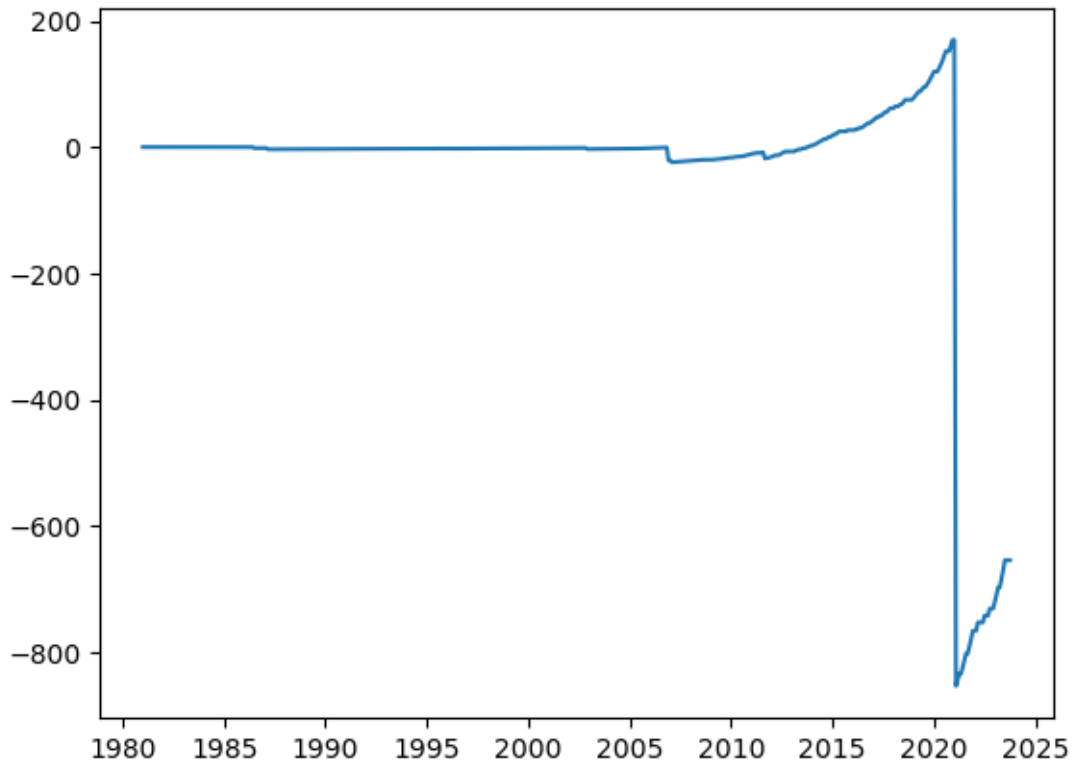
#False Positives
df.loc[(df["Predictions"] == 1) & (df["Target"] == -1), "Profits"] =
    ↪ 100*df['Diff'] + df['Premium']

```

#### 5 5.) Plot profits over time

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

```
[73]: [ <matplotlib.lines.Line2D at 0x1d9ae980c50> ]
```



## 6 5.5) MQE Skills That Could Help Mr. Lui

Mr. Lui's project requires extensive data analysis to optimize ticket prices and create algorithms enhancing user-content creator matching. Advanced statistical and machine learning skills from MQE students would be invaluable. Additionally, incorporating economic interpretability into the data analysis can extract valuable insights from the platform's vast data, potentially generating profits for the company. This multidisciplinary approach can enhance decision-making and strategy formulation.

## 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.

[ ]: