

1.) Pull in Data and Convert ot Monthly

```
In [52]: import yfinance as yf
```

```
In [6]: import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt
```

```
In [26]: apple_data = yf.download('AAPL')  
df = apple_data.resample("M").last()[["Adj Close"]]
```

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

2.) Create columns.

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

```
In [27]: df["Diff"] = df["Adj Close"].diff().shift(-1)  
  
df["Target"] = np.sign(df["Diff"])  
  
df["Premium"] = .08 * df["Adj Close"]
```

```
In [28]: df
```

Out [28]:

	Adj Close	Diff	Target	Premium
Date				
1980-12-31	0.117887	-0.020296	-1.0	0.009431
1981-01-31	0.097591	-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
...
2023-09-30	170.984741	-0.439423	-1.0	13.678779
2023-10-31	170.545319	19.404678	1.0	13.643625
2023-11-30	189.949997	2.580002	1.0	15.196000
2023-12-31	192.529999	-9.850006	-1.0	15.402400
2024-01-31	182.679993	NaN	NaN	14.614399

518 rows x 4 columns

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

```
In [29]: 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
```

```
In [30]: X = pd.read_csv("Xdata.csv", index_col="Date", parse_dates=["Date"])
```

```
In [31]: df
```

Out [31]:

	Adj Close	Diff	Target	Premium
Date				
1980-12-31	0.117887	-0.020296	-1.0	0.009431
1981-01-31	0.097591	-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
...
2023-09-30	170.984741	-0.439423	-1.0	13.678779
2023-10-31	170.545319	19.404678	1.0	13.643625
2023-11-30	189.949997	2.580002	1.0	15.196000
2023-12-31	192.529999	-9.850006	-1.0	15.402400
2024-01-31	182.679993	NaN	NaN	14.614399

518 rows × 4 columns

In [32]: `y = df.loc[:, "2023-09-30", "Target"].copy()`

In [33]: `y`

Out [33]:

Date	
1980-12-31	-1.0
1981-01-31	-1.0
1981-02-28	-1.0
1981-03-31	1.0
1981-04-30	1.0
...	...
2023-05-31	1.0
2023-06-30	1.0
2023-07-31	-1.0
2023-08-31	-1.0
2023-09-30	-1.0

Freq: M, Name: Target, Length: 514, dtype: float64

In [34]: `df=df.loc[:, "2023-09-30", :].copy()`

In [35]: `logreg=LogisticRegression()
logreg.fit(X,y)`

```
Out[35]: ▼ LogisticRegression  
LogisticRegression()
```

4.) Add columns, prediction and profits.

```
In [43]: y_pred=logreg.predict(X)
```

```
In [47]: df["Predictions"]=y_pred
```

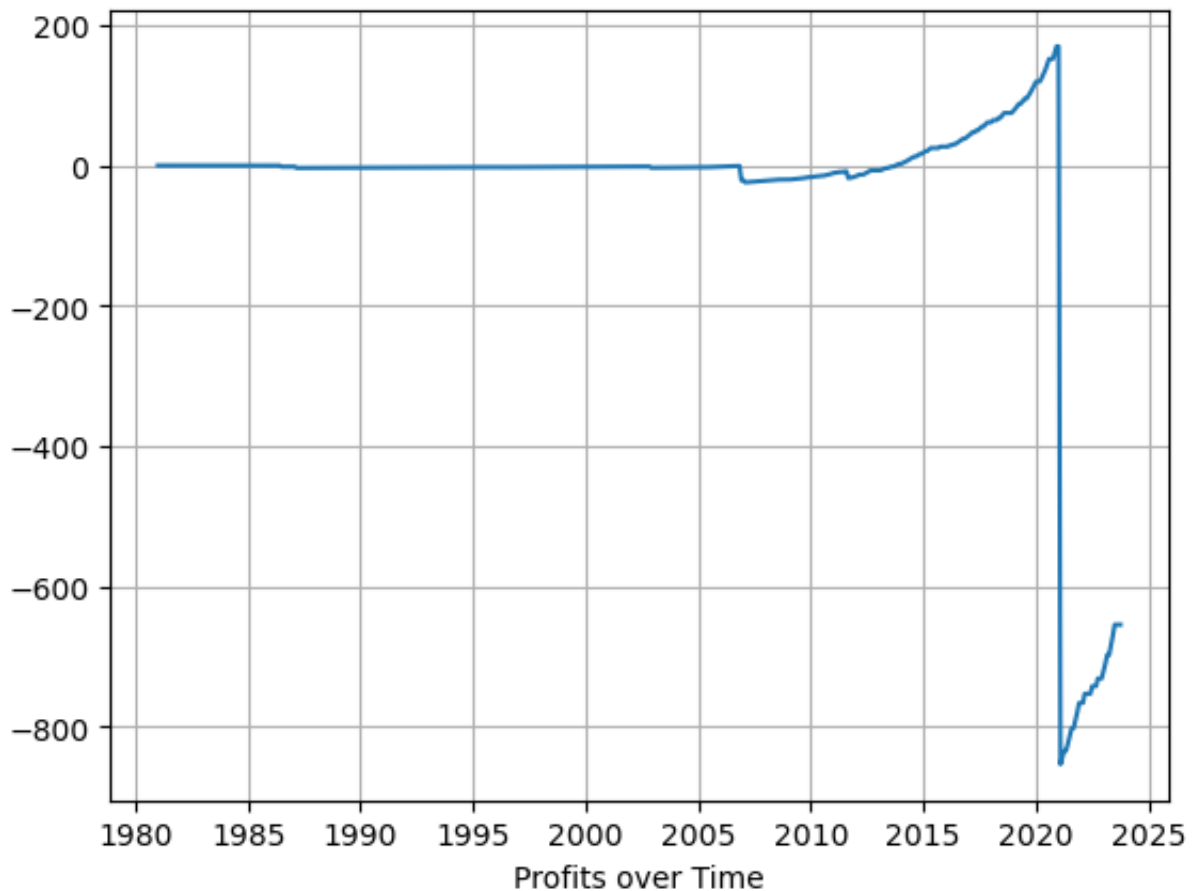
```
In [48]: df["Profits"]=0.
```

```
In [50]: #True Positive Profits  
df.loc[(df["Target"]==1)&(df["Predictions"]==1),"Profits"]=df["Premium"]  
  
df.loc[(df["Target"]==1)&(df["Predictions"]==1),"Profits"]=(100*df["Diff"])
```

5.) Plot profits over time

```
In [51]: plt.plot(np.cumsum(df["Profits"]))  
plt.grid()  
plt.xlabel("Profits over Time")  
plt.plot()
```

```
Out[51]: []
```



```
In [41]: #5.5 how you see ur skills valuable to PJ or Philip Liu

#From Liu's sharing, I realized that our skills combined with
#economics insights and machine learning,
# this could help to reduce the cost of bitcoins such as gas
#fees and improve the ecosystem of bitcoin mining.
```

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

```
In [ ]:
```

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

```
In [ ]:
```