1.) Pull in Data and Convert ot Monthly

In [18]:

```python
import yfinance as yf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [19]:

```python
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 [20]:

```python
# Difference in stockprice
df['Diff']=df["Adj Close"].diff().shift(-1)

# Target up or down
df["Target"]=np.sign(df["Diff"])

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

In [21]:

```python
df.head()
```

Out[21]:

	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

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

In [22]:

```
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 [23]:

```
X = pd.read_csv("/Users/bharatsingh/Downloads/Week 2/Xdata.csv", index_col="
```

In [24]:

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

In [25]:

```
logreg=LogisticRegression()
logreg.fit(X,y)
y_pred=logreg.predict(X)
```

In [26]:

```
df
```

Out[26]:

	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
...	...	...	...	...
2023-05-31	176.778076	16.675476	1.0	14.142246
2023-06-30	193.453552	2.473404	1.0	15.476284
2023-07-31	195.926956	-8.304138	-1.0	15.674156
2023-08-31	187.622818	-16.638077	-1.0	15.009825
2023-09-30	170.984741	-0.439423	-1.0	13.678779

514 rows × 4 columns

4.) Add columns, prediction and profits.

In [27]:

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

In [28]:

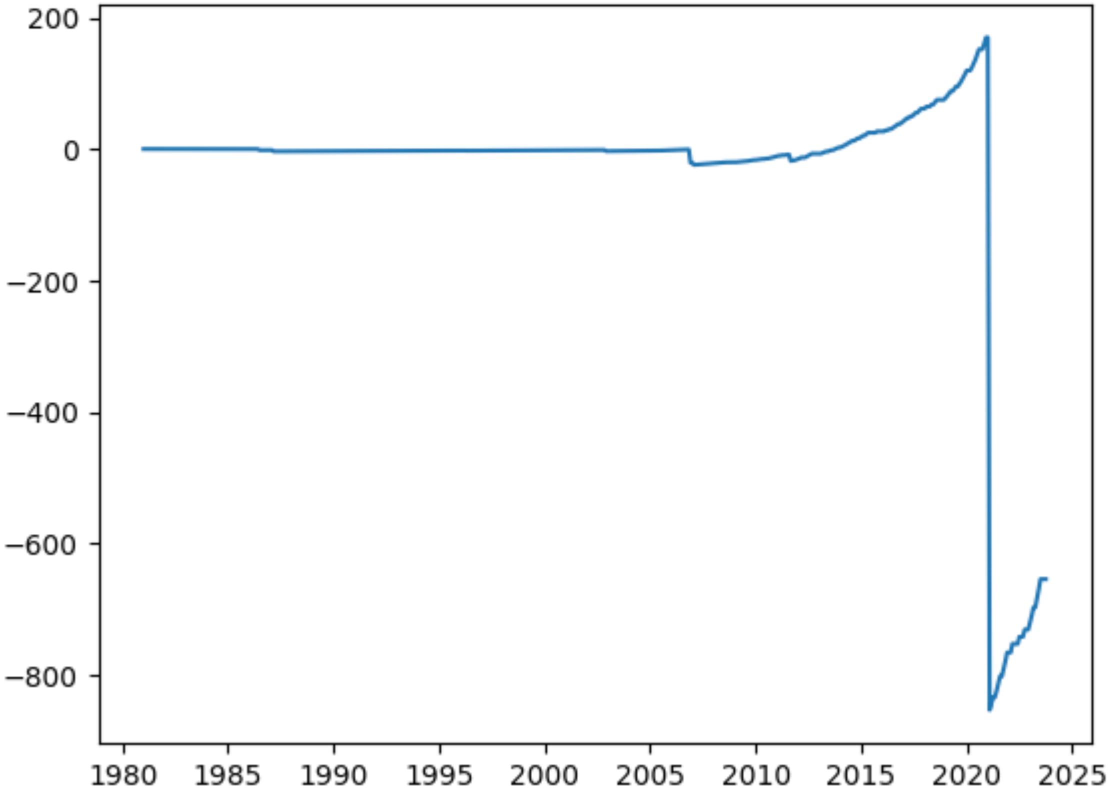
```
df["Profits"]=0

# 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"]+c
```

```
/var/folders/5c/3157mbn910b2gp0sphv4k8xm0000gn/T/ipykernel_41570/357783694
4.py:4: FutureWarning: Setting an item of incompatible dtype is deprecated
and will raise in a future error of pandas. Value '[6.77095473e-03 7.841863
63e-03 4.21455115e-03 5.14728725e-03
 3.52368355e-03 3.73094171e-03 4.97456938e-03 5.04365683e-03
 7.01275647e-03 8.25644493e-03 1.12964559e-02 1.16764355e-02
 1.39564323e-02 9.63824630e-03 5.63095868e-03 6.73641145e-03
 6.84004128e-03 6.84004128e-03 7.04730153e-03 6.84004128e-03
 4.80185032e-03 4.14546281e-03 4.35278654e-03 5.14728725e-03
 5.56187093e-03 6.08002305e-03 6.39097571e-03 6.90912783e-03
 7.80731976e-03 8.36008132e-03 8.63642514e-03 9.25826728e-03
 9.56916094e-03 1.10545897e-02 1.11928284e-02 1.53382862e-02
 1.78255415e-02 2.18661284e-02 2.24196625e-02 2.28348851e-02
 2.99315596e-02 1.83313990e-02 2.30531335e-02 2.22637057e-02
 2.28202486e-02 2.31442952e-02 2.22799540e-02 2.10782647e-02
 2.00128126e-02 2.19087744e-02 2.23790503e-02 2.51126814e-02
 1.92344797e-02 1.92964423e-02 2.23470569e-02 2.34761453e-02
 1.65543365e-02 1.75533032e-02 2.10484362e-02 2.46281385e-02
 3.17874503e-02 3.28587651e-02 2.66131544e-02 2.85475993e-02
 2.93388987e-02 3.25907350e-02 3.74323988e-02 3.37376475e-02
 2.62588167e-02 3.05504060e-02 3.35311675e-02 2.99516225e-02
 1.37496591e-02 1.80878401e-02 1.72673249e-02 1.93335032e-02
 1.57620823e-02 2.00371695e-02 2.01065731e-02 2.22997451e-02
 2.33473849e-02 2.11604738e-02 2.29613757e-02 2.50190616e-02
 2.19184351e-02 1.47587156e-02 1.27151978e-02 1.33207130e-02
 1.34342360e-02 1.39261591e-02 9.83912230e-03 8.62816036e-03
 1.05959940e-02 1.03121924e-02 7.94703007e-03 1.10879242e-02
 1.43046319e-02 1.61210370e-02 1.73698449e-02 1.88835883e-02
 1.93376815e-02 2.47870398e-02 2.10784721e-02 2.17596674e-02
 2.66792011e-02 2.80415249e-02 3.37179852e-02 3.83347893e-02
 4.85145378e-02 5.92619085e-02 6.22514915e-02 6.28191090e-02
 6.94038105e-02 5.08607912e-02 6.15324593e-02 1.80131936e-02
 2.21002340e-02 2.67261219e-02 2.41588759e-02 1.87821448e-02
 2.57936740e-02 2.65202451e-02 2.62780690e-02 2.86637163e-02
 1.75591111e-02 1.73532283e-02 1.73895693e-02 1.71231341e-02
 2.17369533e-02 2.30810928e-02 2.55273008e-02 2.50913215e-02
 2.53214192e-02 2.58784342e-02 2.73194933e-02 2.89664364e-02
 3.12188268e-02 3.39798141e-02 3.91628146e-02 4.17664051e-02
 4.69251108e-02 6.34548616e-02 7.79865217e-02 9.31236744e-02
 8.73352528e-02 8.91517162e-02 1.03295803e-01 1.13564863e-01
 1.29840326e-01 1.39479647e-01 1.64256115e-01 1.74113445e-01
 1.51904211e-01 1.38704596e-01 1.64328804e-01 1.86441097e-01
 1.96371098e-01 2.05477619e-01 2.04920616e-01 2.25022583e-01
 2.41709747e-01 2.93515263e-01 3.19115181e-01 3.35390472e-01
 3.71695404e-01 4.41326256e-01 3.02791271e-01 3.47548637e-01
 4.21296806e-01 3.84967842e-01 2.06712723e-01 2.16303673e-01
 2.54594612e-01 3.04753113e-01 3.44957237e-01 3.95721054e-01
 4.07394943e-01 4.48906975e-01 4.56536102e-01 4.84170418e-01
 4.65158310e-01 4.95577812e-01 5.69156303e-01 6.09191170e-01
 5.88774033e-01 6.87226181e-01 7.28955994e-01 7.53587341e-01
 7.81221542e-01 8.21813202e-01 8.44071274e-01 8.12973175e-01
 9.23534851e-01 9.25666199e-01 9.80886459e-01 1.10556808e+00
 1.31375870e+00 1.39922836e+00 1.41441422e+00 1.47922531e+00
 1.61809052e+00 1.08482460e+00 1.08792137e+00 1.08821609e+00
 9.80998764e-01 1.11954048e+00 1.18724762e+00 1.30167587e+00
 1.39286255e+00 1.25391930e+00 1.32603363e+00 1.35249161e+00
 1.48692444e+00 1.60395889e+00 1.64832764e+00 1.69568649e+00
 1.79592255e+00 1.92515717e+00 1.97611374e+00 2.09749512e+00
 2.23644028e+00 2.24938095e+00 1.99976624e+00 1.77007111e+00
 1.71606659e+00 1.76077240e+00 1.91935226e+00 1.96474808e+00
 2.09344818e+00 2.05710541e+00 2.15575439e+00 2.25868378e+00
 2.56084564e+00 2.68534454e+00 2.70337433e+00 2.79178528e+00
 2.90432037e+00 3.18548065e+00 3.16648376e+00 3.13818695e+00
 3.52868042e+00 3.62742493e+00 3.02801178e+00 3.19501801e+00
 3.33808044e+00 3.66195984e+00 3.38809235e+00 3.83030304e+00
 4.05505005e+00 4.35091248e+00 4.83249146e+00 5.20726990e+00
 5.72166443e+00 4.96650848e+00 5.73817200e+00 6.22645874e+00
 7.14415283e+00 8.32387878e+00 8.54292419e+00 9.35871521e+00
 9.54669189e+00 9.61676086e+00 9.82709717e+00 1.08010535e+01
 1.15029321e+01 1.11758105e+01 1.18313513e+01 1.30746143e+01
 1.30770154e+01 1.08437280e+01 1.09763367e+01 1.03366223e+01
 1.14790369e+01 1.17451563e+01 1.35195581e+01 1.41422461e+01
 1.54762842e+01]' has dtype incompatible with int64, please explicitly cast
to a compatible dtype first.
  df.loc[(df["Predictions"]==1)&(df["Target"]==1),"Profits"]=df["Premium"]
```

5.) Plot profits over time

In [29]:

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



Here, we see that as the false positive pay out significantly less and the detriment is much more than our gain from being correct. Therefore, we have a huge loss during covid.

5.5. My skills from MQE to help MR. Liuz Ventures

By studying the data analysis(python), finance courses(financial engineering), and research techniques I would be able to contribute to Star Arena. It will be done by data collection, visualization on trading activities, interactions on platform which will eventually help the team in decision making. MQE offers the required courses for me to excel in this area and help the platform grow.

```
In [ ]:
```