Kehua Chu (uid: 806153163)

```
In [31]: import pandas as pd
         from sklearn.linear_model import LassoCV
         import matplotlib.pyplot as plt
```

1.) Clean the Apple Data to get a quarterly series of EPS.

```
In [2]: y = pd.read_csv('AAPL_quarterly_financials.csv')
        y.index = y.name
        y = pd.DataFrame(y.loc['BasicEPS', :]).iloc[2:, :]
        y
```

Out[2]:

	BasicEPS
09/30/2023	1.47
06/30/2023	1.27
03/31/2023	1.53
12/31/2022	1.89
09/30/2022	1.29
...	...
09/30/1986	NaN
06/30/1986	0.002
03/31/1986	0.002
12/31/1985	0.004
09/30/1985	NaN

153 rows × 1 columns

```
In [3]: y.index = pd.to_datetime(y.index)
        y = y.sort_index()
        y
```

Out[3]:

	BasicEPS
1985-09-30	NaN
1985-12-31	0.004
1986-03-31	0.002
1986-06-30	0.002
1986-09-30	NaN
...	...
2022-09-30	1.29
2022-12-31	1.89
2023-03-31	1.53
2023-06-30	1.27
2023-09-30	1.47

153 rows × 1 columns

In [4]:

```
y = y.sort_index().fillna(0)
y
```

Out[4]:

	BasicEPS
1985-09-30	0
1985-12-31	0.004
1986-03-31	0.002
1986-06-30	0.002
1986-09-30	0
...	...
2022-09-30	1.29
2022-12-31	1.89
2023-03-31	1.53
2023-06-30	1.27
2023-09-30	1.47

153 rows × 1 columns

2.) Come up with 6 search terms you think could nowcast earnings. (Different than the ones I used) Add in 3 terms that that you think will not Nowcast earnings. Pull in the gtrends data. Clean it to have a quarterly average.

In [49]:

```
# !pip install pytrends
```

```
Collecting pytrends
  Downloading pytrends-4.9.2-py3-none-any.whl (15 kB)
Requirement already satisfied: requests>=2.0 in e:\..kacie\anacondakc\lib\site-packages (from p
ytrends) (2.31.0)
Requirement already satisfied: pandas>=0.25 in e:\..kacie\anacondakc\lib\site-packages (from py
trends) (2.0.3)
Requirement already satisfied: lxml in e:\..kacie\anacondakc\lib\site-packages (from pytrends)
(4.9.3)
Requirement already satisfied: python-dateutil>=2.8.2 in e:\..kacie\anacondakc\lib\site-package
s (from pandas>=0.25->pytrends) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in e:\..kacie\anacondakc\lib\site-packages (from pa
ndas>=0.25->pytrends) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in e:\..kacie\anacondakc\lib\site-packages (from
pandas>=0.25->pytrends) (2023.3)
Requirement already satisfied: numpy>=1.21.0 in e:\..kacie\anacondakc\lib\site-packages (from p
andas>=0.25->pytrends) (1.24.3)
Requirement already satisfied: charset-normalizer<4,>=2 in e:\..kacie\anacondakc\lib\site-packa
ges (from requests>=2.0->pytrends) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in e:\..kacie\anacondakc\lib\site-packages (from re
quests>=2.0->pytrends) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in e:\..kacie\anacondakc\lib\site-packages (f
rom requests>=2.0->pytrends) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in e:\..kacie\anacondakc\lib\site-packages (f
rom requests>=2.0->pytrends) (2023.7.22)
Requirement already satisfied: six>=1.5 in e:\..kacie\anacondakc\lib\site-packages (from python
-dateutil>=2.8.2->pandas>=0.25->pytrends) (1.16.0)
Installing collected packages: pytrends
Successfully installed pytrends-4.9.2
```

In [5]:

```
from pytrends.request import TrendReq
```

In [330…

```
# Create pytrends object
pytrends = TrendReq(hl='en-US', tz=360)

# Set up the keywords and the timeframe
keywords = ['iPhone', 'Samsung', ' Reccession', 'Interest Rates', 'New phone', 'Buy iPhone', 'Se
start_date = '2004-01-01'
end_date = '2024-01-01'

# Create an empty DataFrame to store the results
df = pd.DataFrame()

# Iterate through keywords and fetch data
for keyword in keywords:
    # time.sleep(5) #wrong one
    pytrends.build_payload([keyword], cat=0, timeframe=f'{start_date} {end_date}', geo='', gpro
    interest_over_time_df = pytrends.interest_over_time()
    df[keyword] = interest_over_time_df[keyword]
```

In [331…

```
X = df.resample('Q').mean()
X
```

Out[331]:

date	iPhone	Samsung	Reccession	Interest Rates	New phone	Buy iPhone	Sell iPhone	KPOP tickets	Log	Sun
2004-03-31	0.000000	24.666667	15.333333	60.333333	44.333333	0.000000	0.333333	0.000000	25.000000	39
2004-06-30	0.000000	23.666667	5.333333	67.000000	47.000000	0.333333	0.666667	31.000000	24.666667	53
2004-09-30	0.000000	26.666667	6.000000	52.666667	47.666667	0.000000	0.000000	0.000000	25.000000	41
2004-12-31	0.000000	30.000000	0.000000	46.333333	43.666667	0.333333	0.000000	0.000000	25.000000	33
2005-03-31	0.000000	26.666667	0.000000	47.333333	41.333333	0.000000	0.666667	14.000000	23.333333	32
...	...	...	...	...	...	...	...	...	...	
2023-03-31	50.000000	57.333333	9.666667	88.333333	74.666667	25.000000	25.666667	70.666667	39.333333	34
2023-06-30	43.666667	53.000000	9.000000	74.000000	69.333333	22.000000	22.666667	70.666667	35.666667	50
2023-09-30	52.333333	57.333333	6.333333	74.000000	78.666667	30.333333	31.000000	68.000000	36.333333	56
2023-12-31	51.000000	57.666667	8.000000	71.666667	76.000000	32.333333	30.666667	48.666667	36.000000	25
2024-03-31	49.000000	61.000000	7.000000	79.000000	79.000000	27.000000	29.000000	50.000000	37.000000	27

81 rows × 10 columns

In [332…

```
temp = pd.concat([y, X],axis = 1).dropna()
y = temp[["BasicEPS"]].copy()
X = temp.iloc[:,1:].copy()
```

3.) Normalize all the X data

In [333…

```
from sklearn.preprocessing import StandardScaler
```

In [334…

```
scaler = StandardScaler()
```

In [335…

```
X_scaled = scaler.fit_transform(X)
```

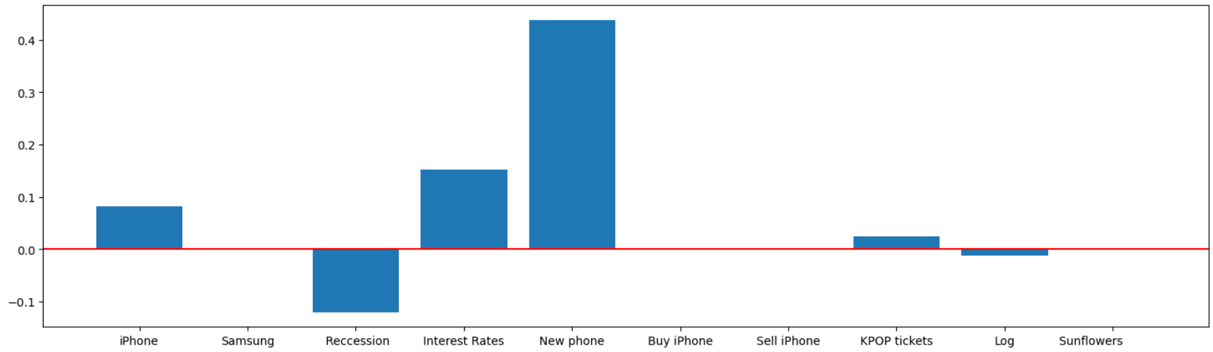
4.) Run a Lasso with lambda of .01. Plot a bar chart.

In [336…

```
from sklearn.linear_model import Lasso
```

In [351…

```
lasso = Lasso(alpha = 0.01)
lasso.fit(X_scaled, y)
coef = lasso.coef_
plt.figure(figsize = (18,5))
plt.bar(range(len(coef)), coef, tick_label = X.columns)
plt.axhline(0, color = 'red')
plt.show()
```



5.) Do these coefficient magnitudes make sense?

Yes, I do think these coefficient magnitudes make sense. As for factors that we think would have significant influence, $iPhone$, $Reccession$, $InterestRates$, $NewPhone$ have the coefficients of relatively larger magnitude, while as for fatocrs we consider not important have the close-to-zero coefficient.

```
In [ ]: # ----------------———————————————————
```