

3 - Lasso, Ridge and Elastic-Net

January 26, 2024

```
[1]: import pandas as pd
```

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

```
[2]: y = pd.read_csv("AAPL_quarterly_financials.csv")
```

```
[3]: y.index = y.name
```

```
[4]: y = pd.DataFrame(y.loc["BasicEPS", :]).iloc[2:,:]
```

```
[5]: y.index = pd.to_datetime(y.index)
```

```
[6]: y = y.sort_index().fillna(0.)
```

2 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

```
[7]: from pytrends.request import TrendReq
```

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

# Set up the keywords and the timeframe
keywords = ['good economy', 'Apple customer', 'inflation', 'game price', '
↳ 'Internet', 'Apple stock', 'clock', 'gasoline', 'bus']

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:
    pytrends.build_payload([keyword], cat=0, timeframe=f'{start_date}_{
    ↪end_date}', geo='', gprop='')
    interest_over_time_df = pytrends.interest_over_time()
    df[keyword] = interest_over_time_df[keyword]
```

```
[9]: X = df.resample("Q").mean()
```

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

3 3.) Normalize all the X data

```
[11]: from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
```

```
[12]: scaler = StandardScaler()
```

```
[13]: X_scaled = scaler.fit_transform(X)
```

4 4.) Run a Lasso with lambda of .5. Plot a bar chart.

```
[14]: from sklearn.linear_model import Lasso
import numpy as np
```

```
[15]: lasso = Lasso(alpha=0.05)
```

```
[16]: lasso.fit(X_scaled, y)
```

```
[16]: Lasso(alpha=0.05)
```

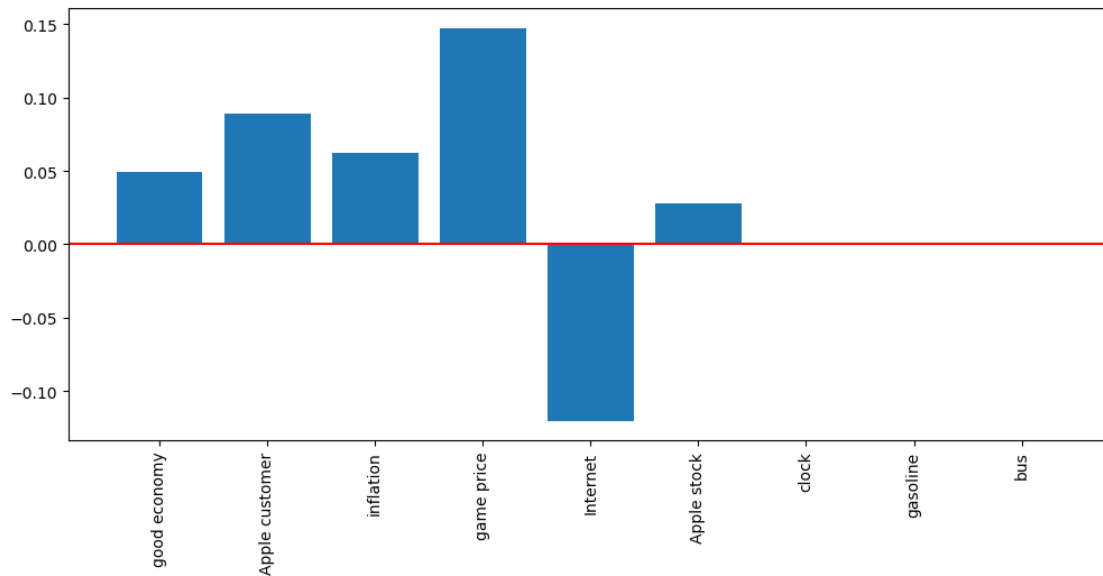
```
[17]: coefficients = lasso.coef_
coefficients
```

```
[17]: array([ 0.04949207,  0.08946733,  0.06207345,  0.14767495, -0.12024969,
            0.02777316, -0.          , -0.          ,  0.          ])
```

```
[18]: import matplotlib.pyplot as plt

plt.figure(figsize=(12, 5))
plt.bar(range(len(coefficients)), coefficients)
plt.xticks(range(len(coefficients)), X.columns, rotation='vertical')
plt.axhline(0, color="red")
```

```
plt.show()
```



```
[19]: lasso1 = Lasso(alpha=0.5)
```

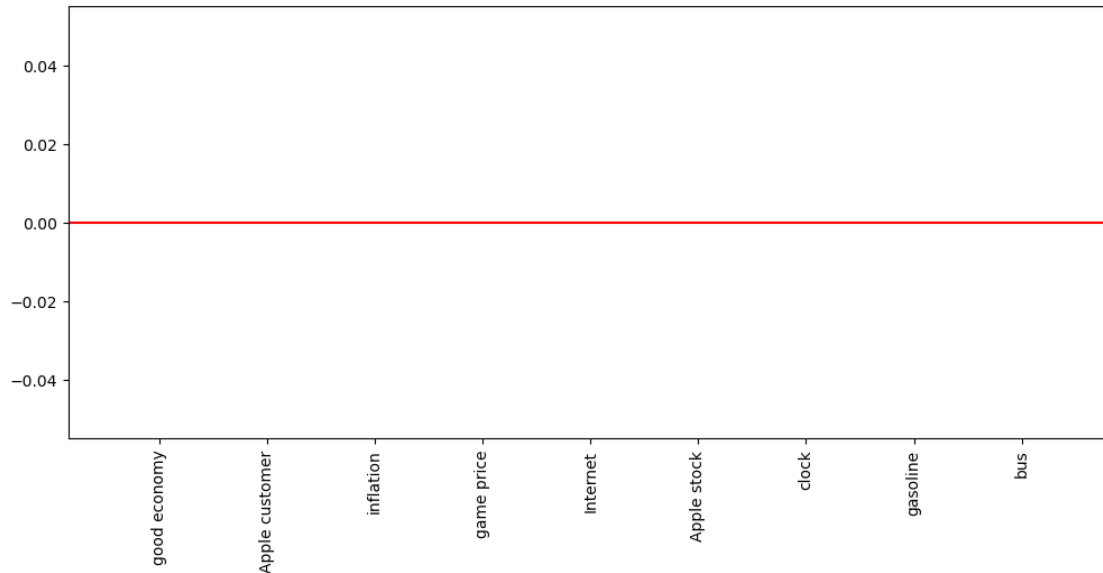
```
[20]: lasso1.fit(X_scaled, y)
```

```
[20]: Lasso(alpha=0.5)
```

```
[21]: coefficients1 = lasso1.coef_  
coefficients1
```

```
[21]: array([ 0.,  0.,  0.,  0., -0.,  0., -0., -0.,  0.])
```

```
[22]: import matplotlib.pyplot as plt  
  
plt.figure(figsize=(12, 5))  
plt.bar(range(len(coefficients1)), coefficients1)  
plt.xticks(range(len(coefficients1)), X.columns, rotation='vertical')  
plt.axhline(0, color="red")  
plt.show()
```



5 5.) Do these coefficient magnitudes make sense?

According to above graphs, if we run a Lasso with a lambda of 0.5, all the coefficients are showing zero, which might be caused by high level of lambda. Then, we run a Lasso with a lambda of 0.05, we can see coefficients.

Yes. These coefficient magnitudes make sense. Variables including good economy, Apple customer, inflation, game price, and Appple stock have positive coefficients, which imply that an increase in these terms is associated with higher EPS. Variable including Internet has negative coefficient since Internet provides more choices, which will negatively affect EPS. In addition, terms such as clock, gasoline, and bus have zero coefficient, indicating they do not contribute to predicting the EPS.