DATA SCIENCE HW4

109511068 | 電機 13 | 崔浩堂

NEWS DATASET

Through the API provided by marketaux, I can update stock market data in real time, so that our model can add stock market forecasts to news elements, making the overall forecast more accurate. Through this API, we will get a Json data in the format of <HTML>, by querying the database within the permission (100 times) every day, we can build a data set of about five years in a month.

As shown in the figure above, we got a data set with stock market data as csv, and determined it through the recent trend of the stock:

1. On the upside, stocks are up

2. Negative, stocks fall

3. Neutral, very affect the stock market

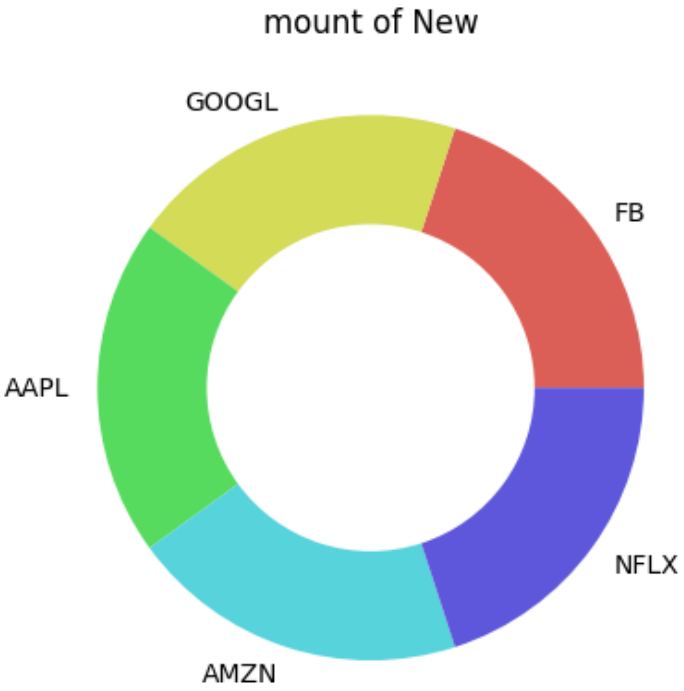
Then, through the weighted score calculation, you can get the compound, and the calculation method is to consider the positive and negative neutral ratio of each stock to get the corresponding value. There are also several columns are the time and news names are also relevant to the stock market.

FAANG_STOCK_NEWS

	ticker	date	time	headline	neg	neu	pos	compound
0	FB	2021-09-22	06:30AM	2 Growth Stocks to Buy Hand Over Fist If the Market Crashes	0.0	0.608	0.392	0.7003
1	FB	2021-09-22	02:23AM	UPDATE 1-Facebook wraps up deals with Australian media firms, TV broadcaster SBS	0.167	0.833	0.0	-0.34
2	FB	2021-09-22	12:36AM	Facebook wraps up deals with Australian media firms, TV broadcaster SBS excluded	0.179	0.821	0.0	-0.34
3	FB	2021-09-22	12:27AM	Facebook wraps up deals with Australia media firms, TV broadcaster SBS not include	0.0	1.0	0.0	0.0
4	FB	2021-09-21	10:34PM	Facebook overpaid FTC fine as quid pro quo to protect Zuckerberg from liability, share	0.099	0.659	0.242	0.3818
5	FB	2021-09-21	06:20PM	Is GBTC Stock A Good Buy As Bitcoin Slumps After Big Rally?	0.0	0.775	0.225	0.4404

PROPORTION OF DATA IN FAANG NEWS

Plotting proportions of a whole might be one of the most common tasks in data visualisation. Examples include regional differences in happiness, economic indicators or crime, demographic differences in voting patterns, income or spending, or contributions of parts of a business to its bottom line. Often, the data also describes changes over time, which may be months, quarters, years or decades.



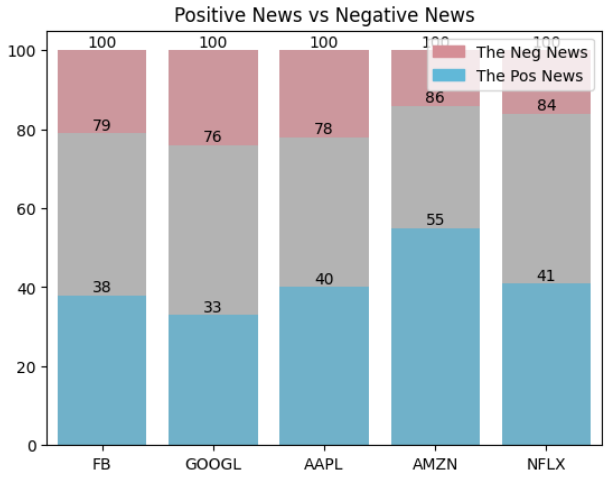
Even though they all relate to proportions of a whole, there often isn't a one-size-fits-all approach that would work for everything.

In this article, I describe what I think are effective techniques for communicating proportions of a whole, and also changes to them over time. I will also explore changes in charts' effectiveness as the number of data points or series change.

```
faang = ["FB", "GOOGL", "AAPL", "AMZN", "NFLX"]
expand = [news[news["ticker"]==n].shape[0] for n in faang]
plt.title("mount of New")
plt.pie(expand, labels = faang, colors = sns.hls_palette(),
        radius=1, wedgeprops={'linewidth':1,'width':0.4})
```

After finding the number of each news through the list constructor, you can use matplotlib's pyplot to draw a pie chart. Because the preset colors are very ugly, I replaced them with the color palette provided by seaborn. Then use the specified radius and the width of the line to hollow out the center of the circle to get a more concise pie chart.

NUMBER OF POSITIVE OR NEGTIVE



In addition to whether each piece of data is the same, the positive and negative scores in the data set should be similar to allow the model to train properly. So I've plotted the following graphs, representing the ratio of positive to negative news for each stock. It can be seen from the figure that they do not have a very even distribution ratio, so I think if you want to train data, using **focal loss** will make them converge faster.

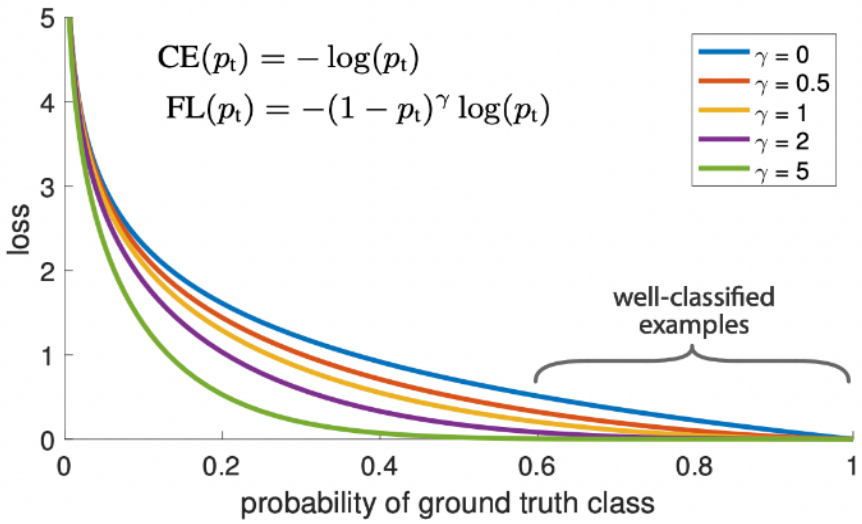
Code implementation:

```
faang = ["FB", "GOOGL", "AAPL", "AMZN", "NFLX"]
state = ["pos", "neg", "neu"]
pos_num, neg_num, neu_num = [], [], []
for stock_name in faang:
    stock_news = news[(news["ticker"] == stock_name)]
    pos_num.append((stock_news["pos"]>stock_news["neg"]).sum())
    neg_num.append((stock_news["pos"]<stock_news["neg"]).sum())
    neu_num.append((stock_news["pos"]==stock_news["neg"]).sum())
neu_sum = [sum(x) for x in zip(pos_num, neu_num)]


import matplotlib.patches as mp
ax=sns.barplot(x=faang,y=[100 for _ in range(5)],color='#d58e96')
ax=sns.barplot(x=faang,y=neu_sum,color='#b3b3b3')
ax=sns.barplot(x=faang,y=pos_num,color='#61b8d8')
for i in range(3): ax.bar_label(ax.containers[i])
blue_patch = mp.Patch(color='#61b8d8', label='The Pos News')
red_patch = mp.Patch(color='#d58e96', label='The Neg News')
gray_patch = mp.Patch(color='#b3b3b3', label='The Neu News')
ax.legend(handles=[red_patch,gray_patch,  blue_patch, ])
ax.set_title('Positive News vs Negative News')
```

FOCAL LOSS

A **Focal Loss** function addresses class imbalance during training in tasks like object detection. Focal loss applies a modulating term to the cross entropy loss in order to focus learning on hard misclassified examples. It is a dynamically scaled cross entropy loss, where the scaling factor decays to zero as confidence in the correct class increases. Intuitively, this scaling factor can automatically down-weight the contribution of easy examples during training and rapidly focus the model on hard examples.



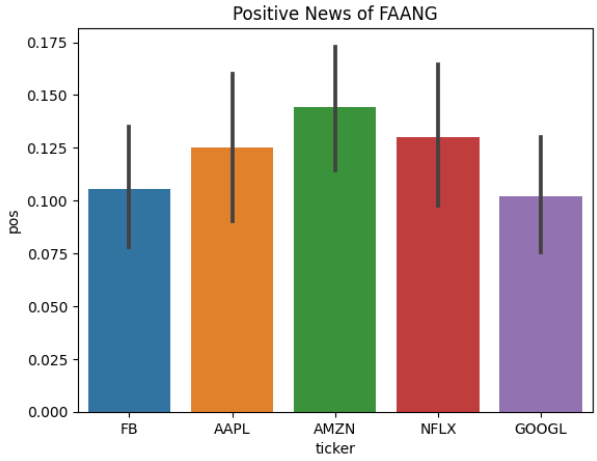
Formally, the Focal Loss adds a factor$(1-p_t)^{\gamma}$ to the standard cross entropy criterion. Setting γ>0 reduces the relative loss for well-classified examples ($p_t > 0.5$), putting more focus on hard, misclassified examples. Here there is tunable *focusing* parameter γ≥0.

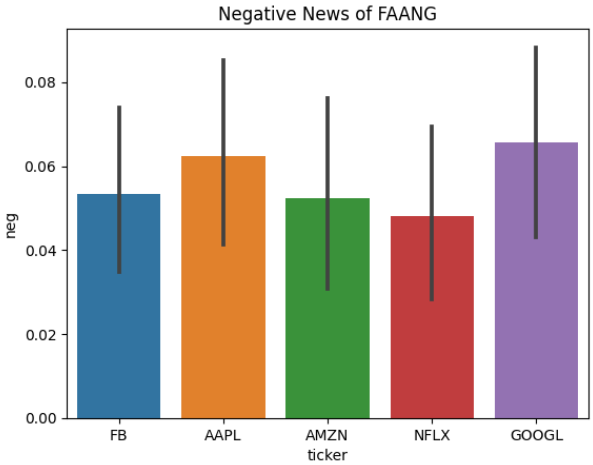
$$FL(p_t) = -(1-p_t)^{\gamma} log(p_t)$$

HOW GOOD IS IT? HOW BAD IS IT?

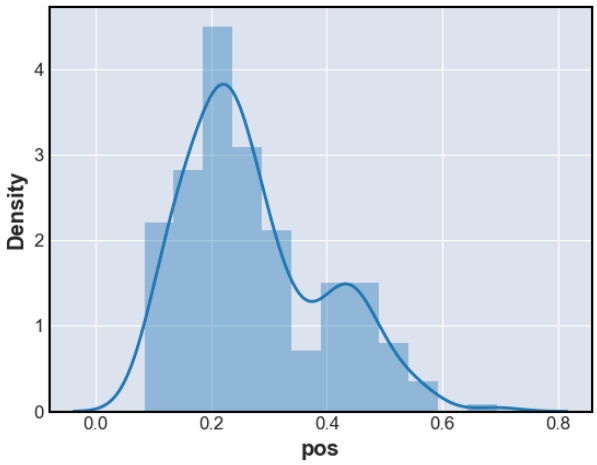
If the model is to be updated today, we must have a significant difference to distinguish. For example, if the bad bad value is only 0.1 today, it should not be estimated that the news is bad. And if the bad value is several times the neutral, it will show their difference more clearly.

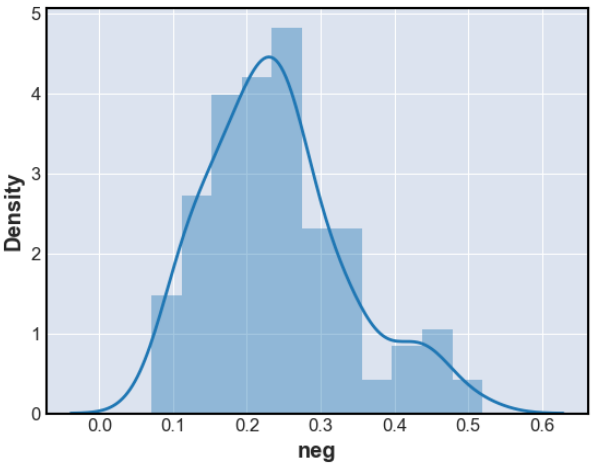
From the chart below, we can observe that the good values are generally relatively large, while the bad values are relatively small, and the standard deviations of the two are almost the same, so we can scale them to mean =1, std = 0.2 at the same time, which can Let the overall value be better analyzed, it is a kind of data argument.





Then we analyze the distribution of the good and bad values of the stocks. Through this distribution, we can know more about how much data can actually be fed back to the model. After all, there is no obvious value, just like neutrality, it cannot give obvious feedback to the model.





According to the plot, we can find that the values are concentrated between 0.2 and 0.3.

I figured I could do an experiment later and compare redistributing all the data to a normal distribution, or training the dataset with the original data. However, the raw data looks like it is not bad, there are not many values close to 0, which is less likely to cause the model to lose accuracy.

Code example:

```
sns.distplot(news["pos"][news["pos"]>0])
```

PRICE OF STOCKS

An OHLC chart is a type of bar chart that shows open, high, low, and closing prices for each period. OHLC charts are useful since they show the four major data points over a period, with the closing price being considered the most important by many traders.

The chart type is useful because it can show increasing or decreasing momentum. When the open and close are far apart it shows strong momentum, and when the open and close are close together it shows indecision or weak momentum. The high and low show the full price range of the period, useful in assessing volatility. There several patterns traders watch for on OHLC charts.

- An OHLC chart shows the open, high, low, and close price for a given period.
- It can be applied to any timeframe.
- The vertical line represents the high and low for the period, while the line to the left marks the open price and the line to the right marks the closing price. This entire structure is called a bar.
- When the close is above the open, the bar is often colored black. When the close is below the open the bar is often colored white.


```
for name in faang:
    data=web.DataReader(name,'yahoo')
    mpf.plot(data,type='candle',mav=(3,6,9),volume=True)
```

Inevitably, we can draw stock information, and we can see that after the Ukrainian-Russian war, all FAANG stocks have dropped significantly. I think all the news at this time will not affect the stock itself, because he has hidden Big news - the Ukrainian-Russian war. Therefore, in the subsequent stock forecasts, the stock market news in the range will be deleted. In addition, we can see that almost all companies have no obvious fluctuations in the near future (within three months), which means that they do not have serious interdependence.

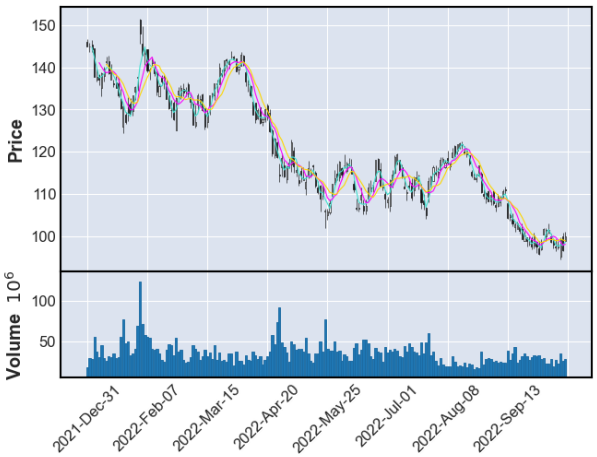
However, you can still consider using the GCN-graph convolution network to analyze the correlation between them.

Since the beginning of the year, FAANG has made a brutal entrance, making many investors question its potential. The market is suffering 16% worse than in 1939. In addition, Netflix has topped the list of wealth destroyers as it keeps hemorrhaging profit and has dropped about 68 percent since January.

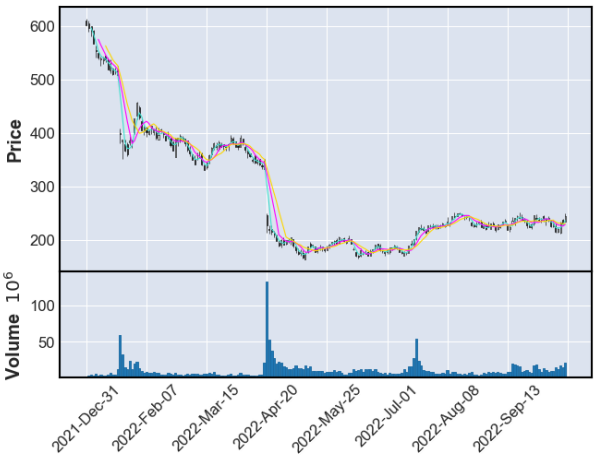
This sudden downfall is leaving many market watchers stunned as the market keeps facing recession fears and rising interest rates, among many others. In addition, tech companies, including FAANG, are experiencing several headwinds.

FB's stock price

GOOGL's stock price



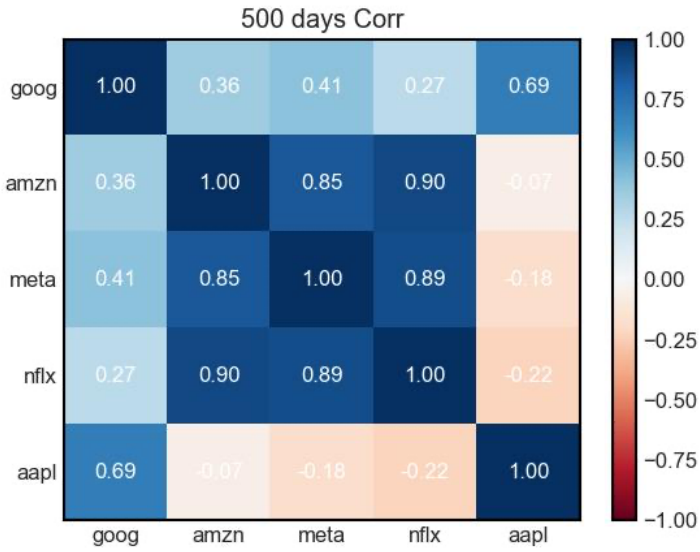
NFLX's stock price



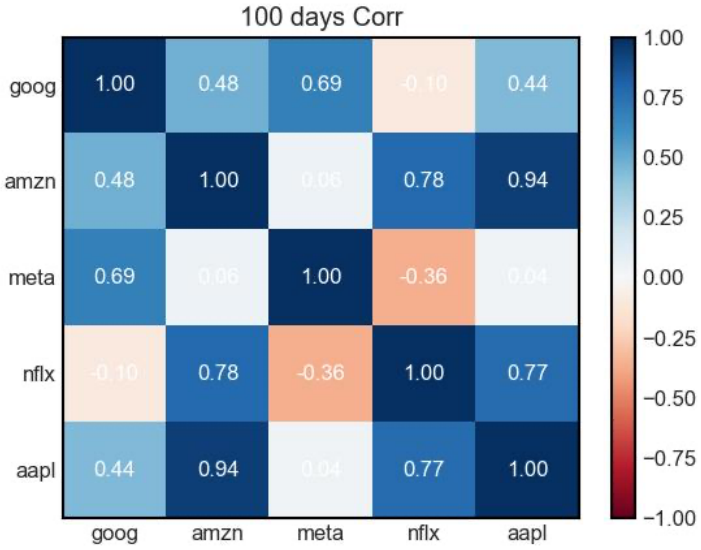
CORRELATION

Similarity refers to whether multiple stocks are related before. We can calculate their relativeness (like hw2) for these five stocks of FAANG, and then visualize the result through custom cmap color.

It can be observed that among the nearly 500-day stocks in the picture above, because of the Ukrainian-Russian war, all stocks have a high similarity. After all, they fell together. However, after that, because AAPL got a successful result at the new machine conference, the stock market price was good. Other companies mostly change stock prices following market volatility and their news.

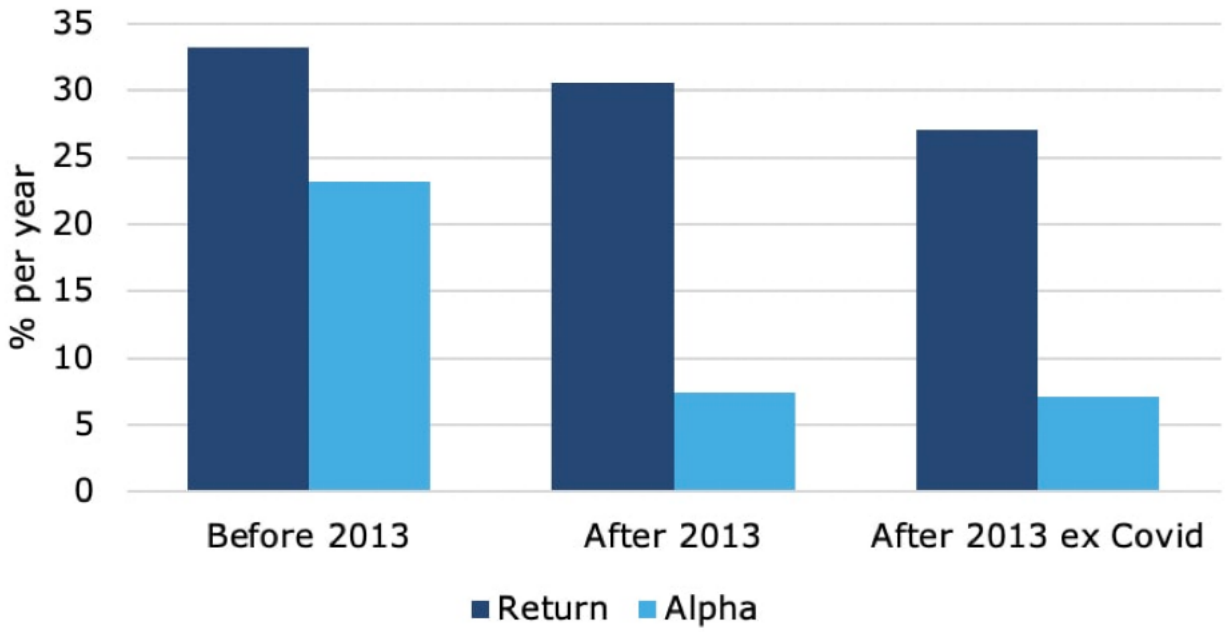


If you put the scale to the recent 100 days, you can see that the correlation is not so limited. At this time, it can indicate whether each company is dependent or opposed. It can be seen from the figure that GOOGL and NFLX are obviously not related, so their correlation is indeed very low, and the two e-commerce companies AMAZ and AAPL have more correlation. This also reflects actual market conditions.



GOOGL

- the raw return of FAANG stocks before they got their acronym in 2013 was about 33% per year. Since 2013, the raw annual return declined to 31%, and if we exclude the Covid period to 27%. That's not so bad it seems.

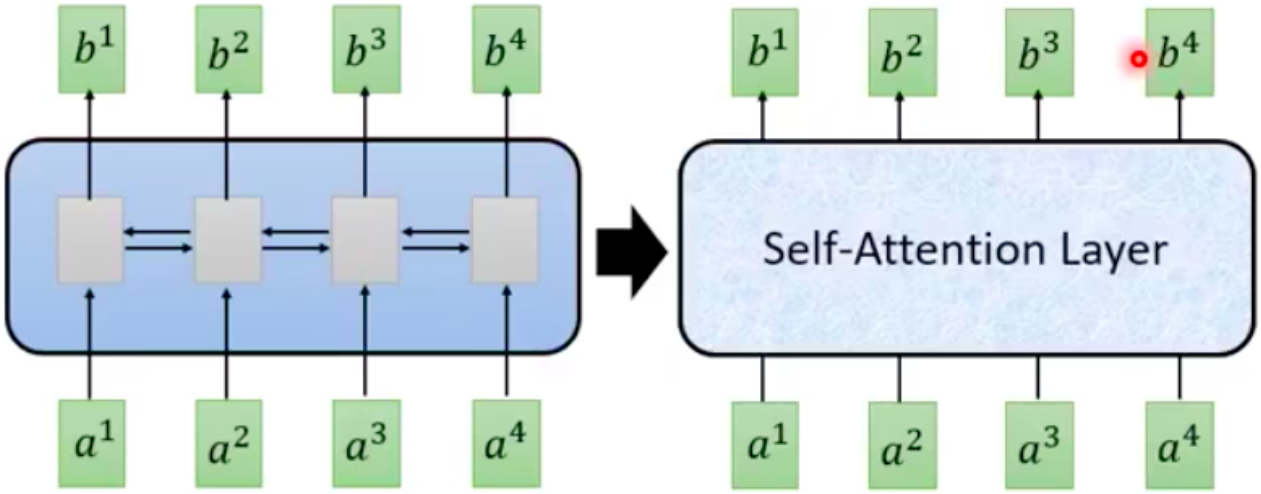


- But if you calculate the alpha of the stocks vs. the stock market and correct for the common four systematic factors market risk, size, value, and momentum, then we see a strong decline. Before 2013, the annual four-factor alpha was some 23% per year. Since then, it has dropped to some 7% per year. That is still a lot, but it is no longer statistically significant. And that implies that going forward, there is no reason to believe that FAANG stocks will outperform the US market. And lest you complain that one problem with this analysis is that FAANG stocks are now so big, they are a major part of the market, the data above for the performance after 2013 is based on alpha vs the market excluding the FAANG stocks, so there is no double counting.

FUTURE WORK

With the above stock market data, we can perform deep learning on the news to analyze whether it will affect the stock market. First of all, because news often has complex headlines, we need to turn it into a vector with features first, which is easier to carry out. analyze. What I thought of is that we can use the self-attention mechanism to encode the news first, and then use the decoder to determine its corresponding value. This is similar to the use of the BERT model, but it will be lighter and not easy. Overfitting on small datasets.

When our dataset gradually becomes larger, we can use models such as Bert to make one-shot predictions. And if there is a chance, I will make up how to use pytorch to do self-attention to the current dataset.



CONCLUSION

In this assignment, I used multiple methods, including pie charts, bar charts and line charts, and even cumulative charts or heat maps to analyze the two existing datasets, so that stock market news and stock prices can be visualized .It is easier to find the relationship between them and predict the difficulties that may be encountered in the future. Makes it easier for later prediction models to avoid weird training problems (such as replacing cross entropy with focal loss). Finally, analyze the factors that may affect the stock market.