



Stock Recommendation System Project Report

CS 6220: Data Mining Techniques

Team Members: Jhalaa Chinoy Tejasvi, Jing Wang, Kit Yu Yip and Qi Zhang

a. Abstract

We plan to build a stock recommendation system by using data mining techniques. The recommendation will be implemented by analyzing the associations between different stocks. If we input one stock symbol, the system will return a group of stocks and show how they are similar to current stock from 3 different perspectives (strong association rules, in the same group and similar price variation trend). In advanced versions, we can input a combination of stocks (may permit to set weight for each stock) and the system will return the results by some methods like weighted average. Additionally, the stock recommendation system could return top 5 stocks with promising trend for people to choose if the input is empty.

b. Introduction

Statement of the problem you are trying to solve

Provide easy and understandable solutions to the users of the system who are new to the stock market and help them make tangible decisions on buying and selling to increase their wealth.

Why is it important to solve this problem?

New investor/ Stock beginner seldom have ideas on how and where to start. While the stock market is a really exciting place to be in, it is also very intimidating. Our stock recommendation system will help such users understand the stock market and provide them with helping hand to make such decisions.

We plan to provide three options to the users to manage their portfolio on the stock market by providing them options to see how similar two stocks are. We are planning to use k-means for this or some other clustering approach.

1. **Strong association rules:** How strongly two stocks are related on basis of their association rules.
2. **In the same group:** If two stocks fall in the same group.
3. **Similar price variation trend:** If two stocks have similar price variation trend.

Background information and bit of literature survey to present what's already known about this problem

The stock market, is very complex and convoluted for a beginner. Political events, market news, quarterly earnings reports, international influence and conflicting trading behaviour, all have a direct impact on how it performs. [7] A lot of studies have shown that predicting stock market returns is a difficult task .[8] A stock is a type of security, which represents ownership in a

company, its assets, earnings and dividends. It is an indicator of how the company is performing and what its growth/profits are. Stocks are traded in stock market, where the prices are controlled by traders' bids (buy price) and offers (sell price). Stocks should be recommended to the investor based on his interests, preferences and trading behaviour. [6] We plan to make this task less time consuming and easily understandable to the user and provide solutions to the same.

c. Methodology

Firstly, we will use Yahoo Finance, Investopedia etc by filtering stock symbols in article content, we can retrieve a list of stocks and also calculate their co-occurrence in the articles. This type of data can be used to generate associations rules.

Then, we will download stock fundamental data like stock category, price, P/E, Volume, market capitalization and EPS et al. Based on these data, we will build a vector model of stock features and use clustering techniques (i.g. k-means) to group them. We can use an "Elbow Curve" to highlight the relationship between how many clusters we choose, and the Sum of Squared Errors (SSE) resulting from using that number of clusters.[5]

Thirdly, we will retrieve historical data of stock prices and analysis the similarity of their variation trend in a period. We will build simple linear regression model for each symbol and calculate the coefficients representing trends. Also, we could use "Stocker" library to gather stock history and predict future stock price change in a certain period. The changed price could as well be the representation of the variation trend. (Stocker ("stock explorer") is a Python-based toolkit for stock analysis and prediction.)[4] The top five promising stock will be generated and saved in this step.

Based on the three types of data obtained above, we will find the best model to predict specific result out of data. Also, we will build a comprehensive recommendation system out of this data.

d. Code

- Brief explanation of the code.

- Retrieve Data

In the codes below, we get all the news links from Investopedia sitemap. We use BeautifulSoup library to parse xml for links.

```

class NewsLinkCrawler:
    # Crawler of page webpage
    def __init__(self, url):
        self.rooturl = url
        self.pageurls = []

    # Get the urls of all news articles
    def run(self):
        html = requests.get(self.rooturl)

        soup = BeautifulSoup(html.text) #Parse webpage by using BeautifulSoup
        links = soup.select("urlset > url > loc")
        for link in links:
            url = link.get_text() #Get the text of each tag
            if '/news/' in url:
                self.pageurls.append(url)
        f = open('newslink.txt', 'w')
        f.write('\n'.join(self.pageurls))
        f.close()

def main():
    url="https://www.investopedia.com/sitemap_1.xml" #Initial url for getting all page tags
    crawler = NewsLinkCrawler(url)
    crawler.run()
main()

```

After running the codes, we can get about 100k news articles. In each article, it includes stock symbols of different companies. For example, in this article(<https://www.investopedia.com/news/twitter-now-using-ai-recommend-tweets-twtr/>), TWTR, FB, GOOG and MSFT are the stock symbols of Twitter, Facebook, Google and Microsoft.

In the codes below (just cut-off screenshot), we retrieve the co-occurrence of stocks in each news article. Similarly, we use BeautifulSoup library to parse the links (pointing to stock page) from article content. Here we have used some regular expression.

After running the code above, we can get the data like this:

```

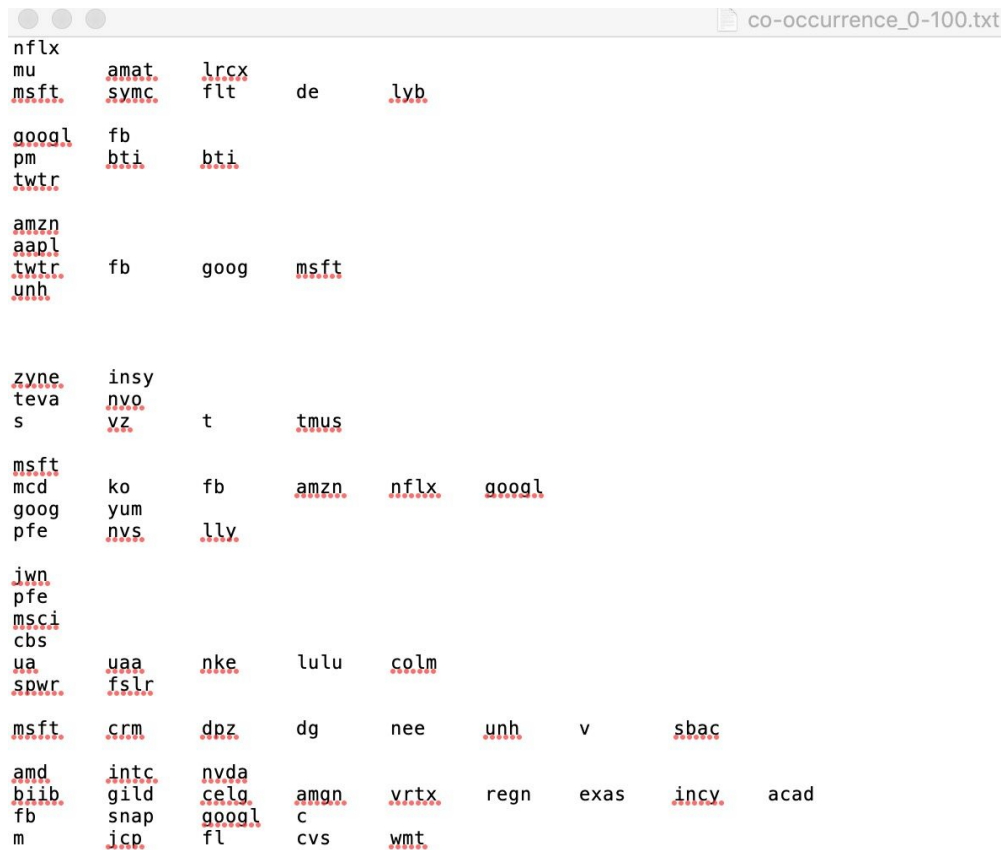
class OccurrenceRetriever:
    # Crawler of page webpage
    def __init__(self, file, start, end):
        self.pageurls = []
        self.occurrences = []
        self.file = file
        self.start = start
        self.end = end

    # Parse stock symbols from news article
    def StockParser(self, url):
        stock_list = []
        html = requests.get(url) #Get the content of page webpage by url
        bsoup = BeautifulSoup(html.text.encode("utf-8"), "xml")
        body_soup = bsoup.find('div', {'class': 'article-body'})
        #print(body_soup)
        for stock in body_soup.findAll('a', {'href': re.compile('https://www.investopedia.com/markets/stocks/*')}):
            matchObj = re.search( r'markets/stocks/(.*)/', stock.get('href'), re.M|re.I)
            if matchObj:
                stock_list.append(matchObj.group(1))
        return stock_list

```

After running the code above, we can get the data like this:

Co-occurrence: twtr, fb, goog, msft



```

nflx
mu
msft.  amat.  lrcx
      symc.  flt    de    lyb

googl  fb
pm     bti    bti
twtr

amzn
aapl
twtr.  fb    goog  msft
unh

zyne
teva
s      insy
      nvda
      vz    t      tmus

msft.  ko     fb    amzn  nflx  googl
mcd    yum
goog   nvs    lly
pfe

jwn
pfe
msci
cbs
ua      uaa    nke    lulu  colm
spwr.  fslr

msft.  crm    dpz    dg     nee    unh    v      sbac

amd    intc   nvda
biib   gild   celg
fb     snap  googl
m      icp    fl     cvs    wmt
  
```

Each line represents a list of stocks in the same article.

- Data Analysis (This part will be completed in the future)

1. Association Rules

According to the data we generate in the previous step, we can use Apriori or FP-growth algorithm to generate items. And then we can get the association rules based on it. The sample codes are: [9]

```
F, support_data = apriori(dataset, min_support=0.04, verbose=True)
```

or

```
F, support_data = fpgrowth(dataset, min_support=0.04, verbose=True)
```

```
H1 = generate_rules(F1, support_data1, min_confidence=0.2, verbose=True)
```

2. K-means Clustering

According to the data we generate in the previous step, we can build K-means clustering to classify different stocks. Then we can get the SSQ statistics or the gap statistics. The sample codes are: [9]

```
ssqs = ssq_statistics (data , ks=range(1, 10+1))
```

```
plot_ssq_statistics(ssqs)
```

or

```
gaps, errs, difs=gap_statistics(data, nrefs=20, ks=range(1, 10+1))  
plot_gap_statistics(gaps, errs, difs)
```

More source codes can be found in the github :

<https://github.com/NUOEL/cs6220>

3. Linear Regression Model

According to the data we generate in the previous step, we can use stocker to build linear regression Model.

e. Results

We will output a list of stocks similar to a given input stock symbol. We also plan to graphically plot these to visually represent how the stocks are similar.

f. Discussion

To be filled later after we publish our results.

g. Future work

To be filled later

- Is this study conclusive or does it lead to some future work? h. conclusion - what conclusions can you draw. i. References – Research papers, articles, and Internet resources referred to in the rest of the report.

Reference

- [1] <https://www.kaggle.com>
- [2] <https://finance.yahoo.com>
- [3] <https://www.investopedia.com>
- [4] <https://github.com/WillKoehrsen/Data-Analysis/blob/master/stocker/Stocker%20Prediction%20Usage.ipynb>
- [5] <https://www.pythonforfinance.net/2018/02/08/stock-clusters-using-k-means-algorithm-in-python/>
- [6] <http://ceur-ws.org/Vol-1606/paper02.pdf>
- [7] <https://www.sciencedirect.com/science/article/pii/S1062976917300443#bib0570>
- [8] <https://www.sciencedirect.com/science/article/pii/S1062976917300443#bib0540>
- [9] <https://github.com/NUOEL/cs6220>