

# HW4-Kmeans programming

Member: Ke Gao

Date: 2019/03/31

Lists of all categories used, and each category list all companies.

Category	Consumer	Finance	Healthcare	Technology
1	AMT	AXP	ABBV	AAPL
2	AMZN	BAC	ABT	ADBE
3	BAM	BK	AMGN	ADP
4	CCI	BLK	ANTM	AVGO
5	CMCSA	C	AZN	CRM
6	COST	CME	BDX	CSCO
7	DIS	COF	BMY	FB
8	HD	GS	CVS	GOOG
9	LOW	HDB	GILD	IBM
10	LVS	ITUB	GSK	INTC
11	MAR	JPM	JNJ	INTU
12	MCD	LFC	LLY	ITW
13	NFLX	MS	MDT	MSFT
14	PLD	PNC	MMM	NVDA
15	RELX	RY	MRK	ORCL
16	SBUX	SAN	NVO	QCOM
17	SPG	SPGI	NVS	SAP
18	TGT	TD	PFE	TSM
19	TJX	USB	SYK	TXN
20	WMT	WFC	UNH	VMW

#### 15 Columns:

['Gross Margin', 'Operating Margin', 'Payout Ratio', 'Tax Rate',  
'Net Margin', 'Asset Turnover', 'Return on Assets', 'Return on Equity',  
'Return on Invested Capital', 'Interest Coverage', 'Current Ratio',  
'Quick Ratio', 'Debt to Equity Ratio', 'Fixed Assets Turnover',  
'Asset Turnover']

### K means result (from 2013 to 2015)

Year = 2013, K = 2

Iteration count =12

Group 0 with 34 element

Ticker list : ['AMZN', 'BAM', 'COST', 'DIS', 'HD', 'LOW', 'MAR', 'NFLX', 'SBUX', 'TGT', 'TJX', 'WMT', 'AXP', 'BAC', 'BK', 'C', 'COF', 'GS', 'HDB', 'ITUB', 'JPM', 'LFC', 'MS', 'PNC', 'RY', 'SAN', 'TD', 'USB', 'WFC', 'ANTM', 'CVS', 'UNH', 'CRM', 'ITW']

Group 1 with 46 element

Ticker list : ['AMT', 'CCI', 'CMCSA', 'LVS', 'MCD', 'PLD', 'RELX', 'SPG', 'BLK', 'CME', 'SPGI', 'ABBV', 'ABT', 'AMGN', 'AZN', 'BDX', 'BMY', 'GILD', 'GSK', 'JNJ', 'LLY', 'MDT', 'MMM', 'MRK', 'NVO', 'NVS', 'PFE', 'SYK', 'AAPL', 'ADBE', 'ADP', 'AVGO', 'CSCO', 'FB', 'GOOG', 'IBM', 'INTC', 'INTU', 'MSFT', 'NVDA', 'ORCL', 'QCOM', 'SAP', 'TSM', 'TXN', 'VMW']

Year = 2013, K = 3

Iteration count =6

Group 0 with 41 element

Ticker list : ['AMT', 'CMCSA', 'LVS', 'MCD', 'PLD', 'RELX', 'SPG', 'SPGI', 'ABBV', 'ABT', 'AMGN', 'AZN', 'BDX', 'BMY', 'GILD', 'GSK', 'JNJ', 'LLY', 'MMM', 'MRK', 'NVO', 'NVS', 'SYK', 'AAPL', 'ADBE', 'ADP', 'AVGO', 'CRM', 'CSCO', 'GOOG', 'IBM', 'INTC', 'INTU', 'ITW', 'MSFT', 'NVDA', 'QCOM', 'SAP', 'TSM', 'TXN', 'VMW']

Group 1 with 32 element

Ticker list : ['AMZN', 'BAM', 'COST', 'DIS', 'HD', 'LOW', 'MAR', 'NFLX', 'SBUX', 'TGT', 'TJX', 'WMT', 'AXP', 'BAC', 'BK', 'C', 'COF', 'GS', 'HDB', 'ITUB', 'JPM', 'LFC', 'MS', 'PNC', 'RY', 'SAN', 'TD', 'USB', 'WFC', 'ANTM', 'CVS', 'UNH']

Group 2 with 7 element

Ticker list : ['CCI', 'BLK', 'CME', 'MDT', 'PFE', 'FB', 'ORCL']

Year = 2013, K = 4

Iteration count =7

Group 0 with 31 element

Ticker list : ['AMT', 'CCI', 'CMCSA', 'DIS', 'LVS', 'MCD', 'PLD', 'RELX', 'SPG', 'ABT', 'AZN', 'BDX', 'BMY', 'GSK', 'JNJ', 'LLY', 'MMM', 'MRK', 'NVS', 'SYK', 'ADBE', 'ADP', 'CRM', 'CSCO', 'IBM', 'INTC', 'ITW', 'NVDA', 'SAP', 'TXN', 'VMW']

Group 1 with 13 element

Ticker list : ['AMZN', 'COST', 'HD', 'LOW', 'MAR', 'NFLX', 'SBUX', 'TGT', 'TJX', 'WMT', 'ANTM', 'CVS', 'UNH']

Group 2 with 18 element

Ticker list : ['BAM', 'AXP', 'BAC', 'BK', 'C', 'COF', 'GS', 'HDB', 'ITUB', 'JPM', 'LFC', 'MS', 'PNC', 'RY', 'SAN', 'TD', 'USB', 'WFC']

Group 3 with 18 element

Ticker list : ['BLK', 'CME', 'SPGI', 'ABBV', 'AMGN', 'GILD', 'MDT', 'NVO', 'PFE', 'AAPL', 'AVGO', 'FB', 'GOOG', 'INTU', 'MSFT', 'ORCL', 'QCOM', 'TSM']

Year = 2013, K = 5

Iteration count =7

Group 0 with 17 element

Ticker list : ['AXP', 'BAC', 'BK', 'C', 'COF', 'GS', 'HDB', 'ITUB', 'JPM', 'LFC', 'MS', 'PNC', 'RY', 'SAN', 'TD', 'USB', 'WFC']

Group 1 with 13 element

Ticker list : ['AMZN', 'COST', 'HD', 'LOW', 'MAR', 'NFLX', 'SBUX', 'TGT', 'TJX', 'WMT', 'ANTM', 'CVS', 'UNH']

Group 2 with 2 element

Ticker list : ['SPGI', 'NVO']

Group 3 with 30 element

Ticker list : ['AMT', 'LVS', 'MCD', 'RELX', 'SPG', 'BLK', 'CME', 'ABBV', 'AMGN', 'GILD', 'GSK', 'JNJ', 'LLY', 'MDT', 'PFE', 'AAPL', 'AVGO', 'CSCO', 'FB', 'GOOG', 'IBM', 'INTC', 'INTU', 'MSFT', 'ORCL', 'QCOM', 'SAP', 'TSM', 'TXN', 'VMW']

Group 4 with 18 element

Ticker list : ['BAM', 'CCI', 'CMCSA', 'DIS', 'PLD', 'ABT', 'AZN', 'BDX', 'BMY', 'MMM', 'MRK', 'NVS', 'SYK', 'ADBE', 'ADP', 'CRM', 'ITW', 'NVDA']

Year = 2013, K = 6

Iteration count =9

Group 0 with 13 element

Ticker list : ['CCI', 'CMCSA', 'PLD', 'ABT', 'AZN', 'BMY', 'MRK', 'NVS', 'SYK', 'ADBE', 'CRM', 'NVDA', 'VMW']

Group 1 with 13 element

Ticker list : ['AMZN', 'COST', 'HD', 'LOW', 'MAR', 'NFLX', 'SBUX', 'TGT', 'TJX', 'WMT', 'ANTM', 'CVS', 'UNH']

Group 2 with 23 element

Ticker list : ['AMT', 'RELX', 'SPG', 'BLK', 'CME', 'ABBV', 'AMGN', 'GILD', 'GSK', 'JNJ', 'LLY', 'MDT', 'PFE', 'AVGO', 'CSCO', 'FB', 'GOOG', 'INTU', 'MSFT', 'ORCL', 'QCOM', 'SAP', 'TSM']

Group 3 with 17 element

Ticker list : ['AXP', 'BAC', 'BK', 'C', 'COF', 'GS', 'HDB', 'ITUB', 'JPM', 'LFC', 'MS', 'PNC', 'RY', 'SAN', 'TD', 'USB', 'WFC']

Group 4 with 12 element

Ticker list : ['BAM', 'DIS', 'LVS', 'MCD', 'BDX', 'MMM', 'AAPL', 'ADP', 'IBM', 'INTC', 'ITW', 'TXN']

Group 5 with 2 element

Ticker list : ['SPGI', 'NVO']

Year = 2013, K = 7

Iteration count =6

Group 0 with 17 element

Ticker list : ['CCI', 'CMCSA', 'PLD', 'ABT', 'AZN', 'BMY', 'JNJ', 'LLY', 'MRK', 'NVS', 'SYK', 'ADBE', 'CRM', 'CSCO', 'NVDA', 'SAP', 'VMW']

Group 1 with 16 element

Ticker list : ['BLK', 'CME', 'AMGN', 'GILD', 'MDT', 'NVO', 'PFE', 'AAPL', 'AVGO', 'FB', 'GOOG', 'INTU', 'MSFT', 'ORCL', 'QCOM', 'TSM']

Group 2 with 9 element

Ticker list : ['DIS', 'LVS', 'MCD', 'BDX', 'MMM', 'ADP', 'INTC', 'ITW', 'TXN']

Group 3 with 13 element

Ticker list : ['AMZN', 'COST', 'HD', 'LOW', 'MAR', 'NFLX', 'SBUX', 'TGT', 'TJX', 'WMT', 'ANTM', 'CVS', 'UNH']

Group 4 with 4 element

Ticker list : ['AMT', 'SPG', 'ABBV', 'GSK']

Group 5 with 18 element

Ticker list : ['BAM', 'AXP', 'BAC', 'BK', 'C', 'COF', 'GS', 'HDB', 'ITUB', 'JPM', 'LFC', 'MS', 'PNC', 'RY', 'SAN', 'TD', 'USB', 'WFC']

Group 6 with 3 element

Ticker list : ['RELX', 'SPGI', 'IBM']

Year = 2013, K = 8

Iteration count =8

Group 0 with 17 element

Ticker list : ['LVS', 'RELX', 'ABT', 'AZN', 'BDX', 'BMY', 'JNJ', 'MRK', 'NVS', 'SYK', 'ADBE', 'CRM', 'CSCO', 'INTC', 'NVDA', 'TXN', 'VMW']

Group 1 with 5 element

Ticker list : ['AMT', 'SPG', 'ABBV', 'GSK', 'IBM']

Group 2 with 1 element

Ticker list : ['SAN']

Group 3 with 16 element

Ticker list : ['AXP', 'BAC', 'BK', 'C', 'COF', 'GS', 'HDB', 'ITUB', 'JPM', 'LFC', 'MS', 'PNC', 'RY', 'TD', 'USB', 'WFC']

Group 4 with 13 element

Ticker list : ['AMZN', 'COST', 'HD', 'LOW', 'MAR', 'NFLX', 'SBUX', 'TGT', 'TJX', 'WMT', 'ANTM', 'CVS', 'UNH']

Group 5 with 9 element

Ticker list : ['BAM', 'CCI', 'CMCSA', 'DIS', 'MCD', 'PLD', 'MMM', 'ADP', 'ITW']

Group 6 with 15 element

Ticker list : ['BLK', 'CME', 'SPGI', 'AMGN', 'GILD', 'LLY', 'MDT', 'PFE', 'FB', 'GOOG', 'INTU', 'MSFT', 'ORCL', 'SAP', 'TSM']

Group 7 with 4 element

Ticker list : ['NVO', 'AAPL', 'AVGO', 'QCOM']

\*\*\*\*\*

Year = 2014, K = 2

Iteration count =10

Group 0 with 30 element

Ticker list : ['AMZN', 'COST', 'HD', 'LOW', 'MAR', 'NFLX', 'TGT', 'WMT', 'AXP', 'BAC', 'BK', 'C', 'COF', 'GS', 'HDB', 'ITUB', 'JPM', 'LFC', 'MS', 'PNC', 'RY', 'SAN', 'SPGI', 'TD', 'USB', 'WFC', 'ANTM', 'CVS', 'UNH', 'CRM']

Group 1 with 50 element

Ticker list : ['AMT', 'BAM', 'CCI', 'CMCSA', 'DIS', 'LVS', 'MCD', 'PLD', 'RELX', 'SBUX', 'SPG', 'TJX', 'BLK', 'CME', 'ABBV', 'ABT', 'AMGN', 'AZN', 'BDX', 'BMY', 'GILD', 'GSK', 'JNJ', 'LLY', 'MDT', 'MMM', 'MRK', 'NVO', 'NVS', 'PFE', 'SYK', 'AAPL', 'ADBE', 'ADP', 'AVGO', 'CSCO', 'FB', 'GOOG', 'IBM', 'INTC', 'INTU', 'ITW', 'MSFT', 'NVDA', 'ORCL', 'QCOM', 'SAP', 'TSM', 'TXN', 'VMW']

Year = 2014, K = 3

Iteration count =9

Group 0 with 41 element

Ticker list : ['AMT', 'BAM', 'CCI', 'CMCSA', 'DIS', 'LVS', 'MCD', 'PLD', 'RELX', 'SBUX', 'SPG', 'CME', 'ABBV', 'ABT', 'AZN', 'BDX', 'BMY', 'GSK', 'JNJ', 'LLY', 'MMM', 'MRK', 'NVO', 'NVS', 'PFE', 'SYK', 'AAPL', 'ADBE', 'ADP', 'AVGO', 'CRM', 'CSCO', 'IBM', 'INTC', 'INTU', 'ITW', 'MSFT', 'SAP', 'TSM', 'TXN', 'VMW']

Group 1 with 9 element

Ticker list : ['BLK', 'AMGN', 'GILD', 'MDT', 'FB', 'GOOG', 'NVDA', 'ORCL', 'QCOM']

Group 2 with 30 element

Ticker list : ['AMZN', 'COST', 'HD', 'LOW', 'MAR', 'NFLX', 'TGT', 'TJX', 'WMT', 'AXP', 'BAC', 'BK', 'C', 'COF', 'GS', 'HDB', 'ITUB', 'JPM', 'LFC', 'MS', 'PNC', 'RY', 'SAN', 'SPGI', 'TD', 'USB', 'WFC', 'ANTM', 'CVS', 'UNH']

Year = 2014, K = 4

Iteration count =11

Group 0 with 17 element

Ticker list : ['AXP', 'BAC', 'BK', 'C', 'COF', 'GS', 'HDB', 'ITUB', 'JPM', 'LFC', 'MS', 'PNC', 'RY', 'SAN', 'TD', 'USB', 'WFC']

Group 1 with 2 element

Ticker list : ['SBUX', 'TJX']

Group 2 with 49 element

Ticker list : ['AMT', 'BAM', 'CCI', 'CMCSA', 'DIS', 'LVS', 'MCD', 'PLD', 'RELX', 'SPG', 'BLK', 'CME', 'ABBV', 'ABT', 'AMGN', 'AZN', 'BDX', 'BMY', 'GILD', 'GSK', 'JNJ', 'LLY', 'MDT', 'MMM', 'MRK', 'NVO', 'NVS', 'PFE', 'SYK', 'AAPL', 'ADBE', 'ADP', 'AVGO', 'CRM', 'CSCO', 'FB', 'GOOG', 'IBM', 'INTC', 'INTU', 'ITW', 'MSFT', 'NVDA', 'ORCL', 'QCOM', 'SAP', 'TSM', 'TXN', 'VMW']

Group 3 with 12 element

Ticker list : ['AMZN', 'COST', 'HD', 'LOW', 'MAR', 'NFLX', 'TGT', 'WMT', 'SPGI', 'ANTM', 'CVS', 'UNH']

Year = 2014, K = 5

Iteration count =5

Group 0 with 15 element

Ticker list : ['BLK', 'AMGN', 'GILD', 'MDT', 'NVO', 'FB', 'GOOG', 'INTU', 'MSFT', 'NVDA', 'ORCL', 'QCOM', 'TSM', 'TXN', 'VMW']

Group 1 with 13 element

Ticker list : ['AMZN', 'COST', 'HD', 'LOW', 'MAR', 'NFLX', 'SBUX', 'TGT', 'TJX', 'WMT', 'ANTM', 'CVS', 'UNH']

Group 2 with 17 element

Ticker list : ['AXP', 'BAC', 'BK', 'C', 'COF', 'GS', 'HDB', 'ITUB', 'JPM', 'LFC', 'MS', 'PNC', 'RY', 'SAN', 'TD', 'USB', 'WFC']

Group 3 with 29 element

Ticker list : ['BAM', 'CMCSA', 'DIS', 'LVS', 'MCD', 'RELX', 'SPGI', 'ABBV', 'ABT', 'BDX', 'BMY', 'GSK', 'JNJ', 'LLY', 'MMM', 'MRK', 'NVS', 'PFE', 'SYK', 'AAPL', 'ADBE', 'ADP', 'AVGO', 'CRM', 'CSCO', 'IBM', 'INTC', 'ITW', 'SAP']

Group 4 with 6 element

Ticker list : ['AMT', 'CCI', 'PLD', 'SPG', 'CME', 'AZN']

Year = 2014, K = 6

Iteration count =8

Group 0 with 12 element

Ticker list : ['CCI', 'CMCSA', 'PLD', 'SPGI', 'ABT', 'AZN', 'BMY', 'LLY', 'SYK', 'ADBE', 'AVGO', 'CRM']

Group 1 with 15 element

Ticker list : ['BLK', 'CME', 'AMGN', 'JNJ', 'MDT', 'PFE', 'CSCO', 'FB', 'GOOG', 'INTU', 'MSFT', 'NVDA', 'ORCL', 'QCOM', 'VMW']

Group 2 with 2 element

Ticker list : ['GILD', 'NVO']

Group 3 with 12 element

Ticker list : ['AMZN', 'COST', 'HD', 'LOW', 'MAR', 'NFLX', 'TGT', 'TJX', 'WMT', 'ANTM', 'CVS', 'UNH']

Group 4 with 22 element

Ticker list : ['AMT', 'BAM', 'DIS', 'LVS', 'MCD', 'RELX', 'SBUX', 'SPG', 'ABBV', 'BDX', 'GSK', 'MMM', 'MRK', 'NVS', 'AAPL', 'ADP', 'IBM', 'INTC', 'ITW', 'SAP', 'TSM', 'TXN']

Group 5 with 17 element

Ticker list : ['AXP', 'BAC', 'BK', 'C', 'COF', 'GS', 'HDB', 'ITUB', 'JPM', 'LFC', 'MS', 'PNC', 'RY', 'SAN', 'TD', 'USB', 'WFC']

Year = 2014, K = 7

Iteration count =6

Group 0 with 21 element

Ticker list : ['AMT', 'DIS', 'LVS', 'MCD', 'RELX', 'SBUX', 'SPG', 'ABBV', 'BDX', 'GSK', 'JNJ', 'MMM', 'MRK', 'AAPL', 'IBM', 'INTC', 'INTU', 'ITW', 'SAP', 'TSM', 'TXN']

Group 1 with 13 element

Ticker list : ['BLK', 'CME', 'AMGN', 'MDT', 'PFE', 'CSCO', 'FB', 'GOOG', 'MSFT', 'NVDA', 'ORCL', 'QCOM', 'VMW']

Group 2 with 17 element

Ticker list : ['AXP', 'BAC', 'BK', 'C', 'COF', 'GS', 'HDB', 'ITUB', 'JPM', 'LFC', 'MS', 'PNC', 'RY', 'SAN', 'TD', 'USB', 'WFC']

Group 3 with 9 element

Ticker list : ['AMZN', 'COST', 'HD', 'LOW', 'MAR', 'TGT', 'TJX', 'WMT', 'CVS']



Group 4 with 2 element  
Ticker list : ['GILD', 'NVO']

Group 5 with 4 element  
Ticker list : ['NFLX', 'SPGI', 'ANTM', 'UNH']

Group 6 with 14 element  
Ticker list : ['BAM', 'CCI', 'CMCSA', 'PLD', 'ABT', 'AZN', 'BMY', 'LLY', 'NVS', 'SYK', 'ADBE', 'ADP', 'AVGO', 'CRM']

Year = 2014, K = 8  
Iteration count = 10  
Group 0 with 14 element  
Ticker list : ['AMT', 'BAM', 'CCI', 'CMCSA', 'SPG', 'ABBV', 'ABT', 'BMY', 'GSK', 'LLY', 'SYK', 'ADBE', 'AVGO', 'CRM']

Group 1 with 5 element  
Ticker list : ['RELX', 'AAPL', 'ADP', 'IBM', 'ITW']

Group 2 with 12 element  
Ticker list : ['AMZN', 'COST', 'HD', 'LOW', 'MAR', 'NFLX', 'TGT', 'TJX', 'WMT', 'ANTM', 'CVS', 'UNH']

Group 3 with 19 element  
Ticker list : ['DIS', 'LVS', 'MCD', 'CME', 'BDX', 'JNJ', 'MDT', 'MMM', 'MRK', 'NVS', 'PFE', 'CSCO', 'INTC', 'INTU', 'MSFT', 'SAP', 'TSM', 'TXN', 'VMW']

Group 4 with 3 element  
Ticker list : ['PLD', 'SBUX', 'AZN']

Group 5 with 1 element  
Ticker list : ['SPGI']

Group 6 with 9 element  
Ticker list : ['BLK', 'AMGN', 'GILD', 'NVO', 'FB', 'GOOG', 'NVDA', 'ORCL', 'QCOM']

Group 7 with 17 element  
Ticker list : ['AXP', 'BAC', 'BK', 'C', 'COF', 'GS', 'HDB', 'ITUB', 'JPM', 'LFC', 'MS', 'PNC', 'RY', 'SAN', 'TD', 'USB', 'WFC']

\*\*\*\*\*

Year = 2015, K = 2  
Iteration count = 10  
Group 0 with 32 element

Ticker list : ['AMZN', 'BAM', 'COST', 'HD', 'LOW', 'MAR', 'NFLX', 'SBUX', 'TGT', 'TJX', 'WMT', 'AXP', 'BAC', 'BK', 'C', 'COF', 'GS', 'HDB', 'ITUB', 'JPM', 'LFC', 'MS', 'PNC', 'RY', 'SAN', 'TD', 'USB', 'WFC', 'ANTM', 'CVS', 'UNH', 'CRM']

Group 1 with 48 element

Ticker list : ['AMT', 'CCI', 'CMCSA', 'DIS', 'LVS', 'MCD', 'PLD', 'RELX', 'SPG', 'BLK', 'CME', 'SPGI', 'ABBV', 'ABT', 'AMGN', 'AZN', 'BDX', 'BMY', 'GILD', 'GSK', 'JNJ', 'LLY', 'MDT', 'MMM', 'MRK', 'NVO', 'NVS', 'PFE', 'SYK', 'AAPL', 'ADBE', 'ADP', 'AVGO', 'CSCO', 'FB', 'GOOG', 'IBM', 'INTC', 'INTU', 'ITW', 'MSFT', 'NVDA', 'ORCL', 'QCOM', 'SAP', 'TSM', 'TXN', 'VMW']

Year = 2015, K = 3

Iteration count =6

Group 0 with 30 element

Ticker list : ['AMZN', 'COST', 'HD', 'LOW', 'MAR', 'NFLX', 'SBUX', 'TGT', 'TJX', 'WMT', 'AXP', 'BAC', 'BK', 'C', 'COF', 'GS', 'HDB', 'ITUB', 'JPM', 'LFC', 'MS', 'PNC', 'RY', 'SAN', 'TD', 'USB', 'WFC', 'ANTM', 'CVS', 'UNH']

Group 1 with 3 element

Ticker list : ['SPGI', 'GILD', 'NVO']

Group 2 with 47 element

Ticker list : ['AMT', 'BAM', 'CCI', 'CMCSA', 'DIS', 'LVS', 'MCD', 'PLD', 'RELX', 'SPG', 'BLK', 'CME', 'ABBV', 'ABT', 'AMGN', 'AZN', 'BDX', 'BMY', 'GSK', 'JNJ', 'LLY', 'MDT', 'MMM', 'MRK', 'NVS', 'PFE', 'SYK', 'AAPL', 'ADBE', 'ADP', 'AVGO', 'CRM', 'CSCO', 'FB', 'GOOG', 'IBM', 'INTC', 'INTU', 'ITW', 'MSFT', 'NVDA', 'ORCL', 'QCOM', 'SAP', 'TSM', 'TXN', 'VMW']

Year = 2015, K = 4

Iteration count =11

Group 0 with 17 element

Ticker list : ['AXP', 'BAC', 'BK', 'C', 'COF', 'GS', 'HDB', 'ITUB', 'JPM', 'LFC', 'MS', 'PNC', 'RY', 'SAN', 'TD', 'USB', 'WFC']

Group 1 with 13 element

Ticker list : ['AMZN', 'COST', 'HD', 'LOW', 'MAR', 'NFLX', 'SBUX', 'TGT', 'TJX', 'WMT', 'ANTM', 'CVS', 'UNH']

Group 2 with 48 element

Ticker list : ['AMT', 'BAM', 'CCI', 'CMCSA', 'DIS', 'LVS', 'MCD', 'PLD', 'RELX', 'SPG', 'BLK', 'CME', 'SPGI', 'ABBV', 'ABT', 'AMGN', 'AZN', 'BDX', 'BMY', 'GSK', 'JNJ', 'LLY', 'MDT', 'MMM', 'MRK', 'NVS', 'PFE', 'SYK', 'AAPL', 'ADBE', 'ADP', 'AVGO', 'CRM', 'CSCO', 'FB', 'GOOG', 'IBM', 'INTC', 'INTU', 'ITW', 'MSFT', 'NVDA', 'ORCL', 'QCOM', 'SAP', 'TSM', 'TXN', 'VMW']

Group 3 with 2 element

Ticker list : ['GILD', 'NVO']

Year = 2015, K = 5

Iteration count =11

Group 0 with 13 element

Ticker list : ['AMT', 'CCI', 'PLD', 'RELX', 'SPG', 'CME', 'SPGI', 'ABBV', 'AZN', 'BMY', 'GSK', 'NVS', 'PFE']

Group 1 with 33 element

Ticker list : ['BAM', 'CMCSA', 'DIS', 'LVS', 'MCD', 'ABT', 'AMGN', 'BDX', 'JNJ', 'LLY', 'MDT', 'MMM', 'MRK', 'SYK', 'AAPL', 'ADBE', 'ADP', 'AVGO', 'CRM', 'CSCO', 'GOOG', 'IBM', 'INTC', 'INTU', 'ITW', 'MSFT', 'NVDA', 'ORCL', 'QCOM', 'SAP', 'TSM', 'TXN', 'VMW']

Group 2 with 4 element

Ticker list : ['BLK', 'GILD', 'NVO', 'FB']

Group 3 with 19 element

Ticker list : ['NFLX', 'AXP', 'BAC', 'BK', 'C', 'COF', 'GS', 'HDB', 'ITUB', 'JPM', 'LFC', 'MS', 'PNC', 'RY', 'SAN', 'TD', 'USB', 'WFC', 'ANTM']

Group 4 with 11 element

Ticker list : ['AMZN', 'COST', 'HD', 'LOW', 'MAR', 'SBUX', 'TGT', 'TJX', 'WMT', 'CVS', 'UNH']

Year = 2015, K = 6

Iteration count =14

Group 0 with 7 element

Ticker list : ['BLK', 'AMGN', 'GILD', 'NVO', 'FB', 'GOOG', 'ORCL']

Group 1 with 11 element

Ticker list : ['AMZN', 'COST', 'HD', 'LOW', 'MAR', 'SBUX', 'TGT', 'TJX', 'WMT', 'CVS', 'UNH']

Group 2 with 21 element

Ticker list : ['AMT', 'CCI', 'CMCSA', 'PLD', 'RELX', 'SPG', 'CME', 'ABBV', 'AZN', 'BMY', 'LLY', 'MDT', 'MRK', 'PFE', 'SYK', 'ADBE', 'CRM', 'INTU', 'MSFT', 'SAP', 'VMW']

Group 3 with 19 element

Ticker list : ['NFLX', 'AXP', 'BAC', 'BK', 'C', 'COF', 'GS', 'HDB', 'ITUB', 'JPM', 'LFC', 'MS', 'PNC', 'RY', 'SAN', 'TD', 'USB', 'WFC', 'ANTM']

Group 4 with 1 element

Ticker list : ['SPGI']

Group 5 with 21 element

Ticker list : ['BAM', 'DIS', 'LVS', 'MCD', 'ABT', 'BDX', 'GSK', 'JNJ', 'MMM', 'NVS', 'AAPL', 'ADP', 'AVGO', 'CSCO', 'IBM', 'INTC', 'ITW', 'NVDA', 'QCOM', 'TSM', 'TXN']

Year = 2015, K = 7

Iteration count =14

Group 0 with 18 element

Ticker list : ['CMCSA', 'BLK', 'AMGN', 'MDT', 'MRK', 'SYK', 'ADBE', 'CRM', 'CSCO', 'FB', 'GOOG', 'INTU', 'MSFT', 'NVDA', 'ORCL', 'QCOM', 'SAP', 'VMW']

Group 1 with 3 element

Ticker list : ['SPGI', 'GILD', 'NVO']

Group 2 with 10 element

Ticker list : ['AMZN', 'COST', 'HD', 'LOW', 'MAR', 'SBUX', 'TGT', 'TJX', 'WMT', 'CVS']

Group 3 with 11 element

Ticker list : ['AMT', 'CCI', 'PLD', 'SPG', 'CME', 'ABBV', 'AZN', 'BMY', 'LLY', 'NVS', 'PFE']

Group 4 with 3 element

Ticker list : ['NFLX', 'ANTM', 'UNH']

Group 5 with 17 element

Ticker list : ['AXP', 'BAC', 'BK', 'C', 'COF', 'GS', 'HDB', 'ITUB', 'JPM', 'LFC', 'MS', 'PNC', 'RY', 'SAN', 'TD', 'USB', 'WFC']

Group 6 with 18 element

Ticker list : ['BAM', 'DIS', 'LVS', 'MCD', 'RELX', 'ABT', 'BDX', 'GSK', 'JNJ', 'MMM', 'AAPL', 'ADP', 'AVGO', 'IBM', 'INTC', 'ITW', 'TSM', 'TXN']

Year = 2015, K = 8

Iteration count =12

Group 0 with 4 element

Ticker list : ['SPGI', 'ABBV', 'GILD', 'GSK']

Group 1 with 21 element

Ticker list : ['AMT', 'CCI', 'CMCSA', 'PLD', 'RELX', 'SPG', 'CME', 'AZN', 'BMY', 'JNJ', 'LLY', 'MDT', 'MRK', 'NVS', 'PFE', 'SYK', 'ADBE', 'INTU', 'MSFT', 'SAP', 'VMW']

Group 2 with 1 element

Ticker list : ['NVO']

Group 3 with 1 element

Ticker list : ['CRM']

Group 4 with 11 element

Ticker list : ['AMZN', 'COST', 'HD', 'LOW', 'MAR', 'SBUX', 'TGT', 'TJX', 'WMT', 'CVS', 'UNH']

Group 5 with 5 element

Ticker list : ['BLK', 'AMGN', 'FB', 'GOOG', 'ORCL']

Group 6 with 19 element

Ticker list : ['NFLX', 'AXP', 'BAC', 'BK', 'C', 'COF', 'GS', 'HDB', 'ITUB', 'JPM', 'LFC', 'MS', 'PNC', 'RY', 'SAN', 'TD', 'USB', 'WFC', 'ANTM']

Group 7 with 18 element

Ticker list : ['BAM', 'DIS', 'LVS', 'MCD', 'ABT', 'BDX', 'MMM', 'AAPL', 'ADP', 'AVGO', 'CSCO', 'IBM', 'INTC', 'ITW', 'NVDA', 'QCOM', 'TSM', 'TXN']

\*\*\*\*\*

**Companies that with 80% or more similarity.**

[ AMT , SPG ] Similarity = 1.0  
[ AMT , ABBV ] Similarity = 0.8571428571428571  
[ AMT , GSK ] Similarity = 0.8095238095238095  
[ AMZN , COST ] Similarity = 1.0  
[ AMZN , HD ] Similarity = 1.0  
[ AMZN , LOW ] Similarity = 1.0  
[ AMZN , MAR ] Similarity = 1.0  
[ AMZN , TGT ] Similarity = 1.0  
[ AMZN , TJX ] Similarity = 0.9047619047619048  
[ AMZN , WMT ] Similarity = 1.0  
[ AMZN , CVS ] Similarity = 1.0  
[ AMZN , UNH ] Similarity = 0.9047619047619048  
[ CCI , CMCSA ] Similarity = 0.8095238095238095  
[ CCI , PLD ] Similarity = 0.9047619047619048  
[ CCI , AZN ] Similarity = 0.8571428571428571  
[ CCI , BMY ] Similarity = 0.8571428571428571  
[ CMCSA , PLD ] Similarity = 0.8095238095238095  
[ CMCSA , ABT ] Similarity = 0.8095238095238095  
[ CMCSA , BMY ] Similarity = 0.8571428571428571  
[ CMCSA , LLY ] Similarity = 0.8095238095238095  
[ CMCSA , MRK ] Similarity = 0.8095238095238095  
[ CMCSA , SYK ] Similarity = 0.9523809523809523  
[ CMCSA , ADBE ] Similarity = 0.9523809523809523  
[ COST , HD ] Similarity = 1.0  
[ COST , LOW ] Similarity = 1.0  
[ COST , MAR ] Similarity = 1.0  
[ COST , TGT ] Similarity = 1.0  
[ COST , TJX ] Similarity = 0.9047619047619048  
[ COST , WMT ] Similarity = 1.0  
[ COST , CVS ] Similarity = 1.0  
[ COST , UNH ] Similarity = 0.9047619047619048  
[ DIS , LVS ] Similarity = 0.8095238095238095  
[ DIS , MCD ] Similarity = 0.8571428571428571  
[ DIS , BDX ] Similarity = 0.8571428571428571  
[ DIS , MMM ] Similarity = 0.9047619047619048  
[ DIS , ADP ] Similarity = 0.8095238095238095  
[ DIS , INTC ] Similarity = 0.8095238095238095  
[ DIS , ITW ] Similarity = 0.9047619047619048  
[ HD , LOW ] Similarity = 1.0  
[ HD , MAR ] Similarity = 1.0  
[ HD , TGT ] Similarity = 1.0  
[ HD , TJX ] Similarity = 0.9047619047619048  
[ HD , WMT ] Similarity = 1.0

[ HD , CVS ] Similarity = 1.0  
[ HD , UNH ] Similarity = 0.9047619047619048  
[ LOW , MAR ] Similarity = 1.0  
[ LOW , TGT ] Similarity = 1.0  
[ LOW , TJX ] Similarity = 0.9047619047619048  
[ LOW , WMT ] Similarity = 1.0  
[ LOW , CVS ] Similarity = 1.0  
[ LOW , UNH ] Similarity = 0.9047619047619048  
[ LVS , MCD ] Similarity = 0.9523809523809523  
[ LVS , BDX ] Similarity = 0.9523809523809523  
[ LVS , JNJ ] Similarity = 0.8095238095238095  
[ LVS , MMM ] Similarity = 0.9047619047619048  
[ LVS , AAPL ] Similarity = 0.8095238095238095  
[ LVS , ADP ] Similarity = 0.8095238095238095  
[ LVS , IBM ] Similarity = 0.8571428571428571  
[ LVS , INTC ] Similarity = 1.0  
[ LVS , ITW ] Similarity = 0.8095238095238095  
[ LVS , TXN ] Similarity = 0.9523809523809523  
[ MAR , TGT ] Similarity = 1.0  
[ MAR , TJX ] Similarity = 0.9047619047619048  
[ MAR , WMT ] Similarity = 1.0  
[ MAR , CVS ] Similarity = 1.0  
[ MAR , UNH ] Similarity = 0.9047619047619048  
[ MCD , BDX ] Similarity = 0.9047619047619048  
[ MCD , MMM ] Similarity = 0.9523809523809523  
[ MCD , AAPL ] Similarity = 0.8095238095238095  
[ MCD , ADP ] Similarity = 0.8571428571428571  
[ MCD , IBM ] Similarity = 0.8571428571428571  
[ MCD , INTC ] Similarity = 0.9523809523809523  
[ MCD , ITW ] Similarity = 0.8571428571428571  
[ MCD , TXN ] Similarity = 0.9047619047619048  
[ NFLX , ANTM ] Similarity = 1.0  
[ NFLX , UNH ] Similarity = 0.8571428571428571  
[ PLD , AZN ] Similarity = 0.9523809523809523  
[ PLD , BMY ] Similarity = 0.8571428571428571  
[ SBUX , TJX ] Similarity = 0.8095238095238095  
[ SPG , ABBV ] Similarity = 0.8571428571428571  
[ SPG , GSK ] Similarity = 0.8095238095238095  
[ TGT , TJX ] Similarity = 0.9047619047619048  
[ TGT , WMT ] Similarity = 1.0  
[ TGT , CVS ] Similarity = 1.0  
[ TGT , UNH ] Similarity = 0.9047619047619048  
[ TJX , WMT ] Similarity = 0.9047619047619048  
[ TJX , CVS ] Similarity = 0.9047619047619048

[ TJX , UNH ] Similarity = 0.8095238095238095  
[ WMT , CVS ] Similarity = 1.0  
[ WMT , UNH ] Similarity = 0.9047619047619048  
[ AXP , BAC ] Similarity = 1.0  
[ AXP , BK ] Similarity = 1.0  
[ AXP , C ] Similarity = 1.0  
[ AXP , COF ] Similarity = 1.0  
[ AXP , GS ] Similarity = 1.0  
[ AXP , HDB ] Similarity = 1.0  
[ AXP , ITUB ] Similarity = 1.0  
[ AXP , JPM ] Similarity = 1.0  
[ AXP , LFC ] Similarity = 1.0  
[ AXP , MS ] Similarity = 1.0  
[ AXP , PNC ] Similarity = 1.0  
[ AXP , RY ] Similarity = 1.0  
[ AXP , SAN ] Similarity = 0.9523809523809523  
[ AXP , TD ] Similarity = 1.0  
[ AXP , USB ] Similarity = 1.0  
[ AXP , WFC ] Similarity = 1.0  
[ BAC , BK ] Similarity = 1.0  
[ BAC , C ] Similarity = 1.0  
[ BAC , COF ] Similarity = 1.0  
[ BAC , GS ] Similarity = 1.0  
[ BAC , HDB ] Similarity = 1.0  
[ BAC , ITUB ] Similarity = 1.0  
[ BAC , JPM ] Similarity = 1.0  
[ BAC , LFC ] Similarity = 1.0  
[ BAC , MS ] Similarity = 1.0  
[ BAC , PNC ] Similarity = 1.0  
[ BAC , RY ] Similarity = 1.0  
[ BAC , SAN ] Similarity = 0.9523809523809523  
[ BAC , TD ] Similarity = 1.0  
[ BAC , USB ] Similarity = 1.0  
[ BAC , WFC ] Similarity = 1.0  
[ BK , C ] Similarity = 1.0  
[ BK , COF ] Similarity = 1.0  
[ BK , GS ] Similarity = 1.0  
[ BK , HDB ] Similarity = 1.0  
[ BK , ITUB ] Similarity = 1.0  
[ BK , JPM ] Similarity = 1.0  
[ BK , LFC ] Similarity = 1.0  
[ BK , MS ] Similarity = 1.0  
[ BK , PNC ] Similarity = 1.0  
[ BK , RY ] Similarity = 1.0



[ BK , SAN ] Similarity = 0.9523809523809523  
[ BK , TD ] Similarity = 1.0  
[ BK , USB ] Similarity = 1.0  
[ BK , WFC ] Similarity = 1.0  
[ BLK , AMGN ] Similarity = 0.9047619047619048  
[ BLK , MDT ] Similarity = 0.8095238095238095  
[ BLK , FB ] Similarity = 1.0  
[ BLK , GOOG ] Similarity = 0.9047619047619048  
[ BLK , ORCL ] Similarity = 0.9523809523809523  
[ C , COF ] Similarity = 1.0  
[ C , GS ] Similarity = 1.0  
[ C , HDB ] Similarity = 1.0  
[ C , ITUB ] Similarity = 1.0  
[ C , JPM ] Similarity = 1.0  
[ C , LFC ] Similarity = 1.0  
[ C , MS ] Similarity = 1.0  
[ C , PNC ] Similarity = 1.0  
[ C , RY ] Similarity = 1.0  
[ C , SAN ] Similarity = 0.9523809523809523  
[ C , TD ] Similarity = 1.0  
[ C , USB ] Similarity = 1.0  
[ C , WFC ] Similarity = 1.0  
[ CME , MDT ] Similarity = 0.8095238095238095  
[ CME , PFE ] Similarity = 0.9523809523809523  
[ CME , MSFT ] Similarity = 0.8095238095238095  
[ COF , GS ] Similarity = 1.0  
[ COF , HDB ] Similarity = 1.0  
[ COF , ITUB ] Similarity = 1.0  
[ COF , JPM ] Similarity = 1.0  
[ COF , LFC ] Similarity = 1.0  
[ COF , MS ] Similarity = 1.0  
[ COF , PNC ] Similarity = 1.0  
[ COF , RY ] Similarity = 1.0  
[ COF , SAN ] Similarity = 0.9523809523809523  
[ COF , TD ] Similarity = 1.0  
[ COF , USB ] Similarity = 1.0  
[ COF , WFC ] Similarity = 1.0  
[ GS , HDB ] Similarity = 1.0  
[ GS , ITUB ] Similarity = 1.0  
[ GS , JPM ] Similarity = 1.0  
[ GS , LFC ] Similarity = 1.0  
[ GS , MS ] Similarity = 1.0  
[ GS , PNC ] Similarity = 1.0  
[ GS , RY ] Similarity = 1.0

[ GS , SAN ] Similarity = 0.9523809523809523  
[ GS , TD ] Similarity = 1.0  
[ GS , USB ] Similarity = 1.0  
[ GS , WFC ] Similarity = 1.0  
[ HDB , ITUB ] Similarity = 1.0  
[ HDB , JPM ] Similarity = 1.0  
[ HDB , LFC ] Similarity = 1.0  
[ HDB , MS ] Similarity = 1.0  
[ HDB , PNC ] Similarity = 1.0  
[ HDB , RY ] Similarity = 1.0  
[ HDB , SAN ] Similarity = 0.9523809523809523  
[ HDB , TD ] Similarity = 1.0  
[ HDB , USB ] Similarity = 1.0  
[ HDB , WFC ] Similarity = 1.0  
[ ITUB , JPM ] Similarity = 1.0  
[ ITUB , LFC ] Similarity = 1.0  
[ ITUB , MS ] Similarity = 1.0  
[ ITUB , PNC ] Similarity = 1.0  
[ ITUB , RY ] Similarity = 1.0  
[ ITUB , SAN ] Similarity = 0.9523809523809523  
[ ITUB , TD ] Similarity = 1.0  
[ ITUB , USB ] Similarity = 1.0  
[ ITUB , WFC ] Similarity = 1.0  
[ JPM , LFC ] Similarity = 1.0  
[ JPM , MS ] Similarity = 1.0  
[ JPM , PNC ] Similarity = 1.0  
[ JPM , RY ] Similarity = 1.0  
[ JPM , SAN ] Similarity = 0.9523809523809523  
[ JPM , TD ] Similarity = 1.0  
[ JPM , USB ] Similarity = 1.0  
[ JPM , WFC ] Similarity = 1.0  
[ LFC , MS ] Similarity = 1.0  
[ LFC , PNC ] Similarity = 1.0  
[ LFC , RY ] Similarity = 1.0  
[ LFC , SAN ] Similarity = 0.9523809523809523  
[ LFC , TD ] Similarity = 1.0  
[ LFC , USB ] Similarity = 1.0  
[ LFC , WFC ] Similarity = 1.0  
[ MS , PNC ] Similarity = 1.0  
[ MS , RY ] Similarity = 1.0  
[ MS , SAN ] Similarity = 0.9523809523809523  
[ MS , TD ] Similarity = 1.0  
[ MS , USB ] Similarity = 1.0  
[ MS , WFC ] Similarity = 1.0

[ PNC , RY ] Similarity = 1.0  
[ PNC , SAN ] Similarity = 0.9523809523809523  
[ PNC , TD ] Similarity = 1.0  
[ PNC , USB ] Similarity = 1.0  
[ PNC , WFC ] Similarity = 1.0  
[ RY , SAN ] Similarity = 0.9523809523809523  
[ RY , TD ] Similarity = 1.0  
[ RY , USB ] Similarity = 1.0  
[ RY , WFC ] Similarity = 1.0  
[ SAN , TD ] Similarity = 0.9523809523809523  
[ SAN , USB ] Similarity = 0.9523809523809523  
[ SAN , WFC ] Similarity = 0.9523809523809523  
[ TD , USB ] Similarity = 1.0  
[ TD , WFC ] Similarity = 1.0  
[ USB , WFC ] Similarity = 1.0  
[ ABBV , GSK ] Similarity = 0.8571428571428571  
[ ABT , BMY ] Similarity = 0.8095238095238095  
[ ABT , SYK ] Similarity = 0.8571428571428571  
[ ABT , ADBE ] Similarity = 0.8571428571428571  
[ AMGN , MDT ] Similarity = 0.8095238095238095  
[ AMGN , FB ] Similarity = 0.9047619047619048  
[ AMGN , GOOG ] Similarity = 1.0  
[ AMGN , MSFT ] Similarity = 0.8095238095238095  
[ AMGN , ORCL ] Similarity = 0.9523809523809523  
[ AMGN , QCOM ] Similarity = 0.8571428571428571  
[ ANTM , UNH ] Similarity = 0.8571428571428571  
[ AZN , BMY ] Similarity = 0.9047619047619048  
[ AZN , NVS ] Similarity = 0.8095238095238095  
[ AZN , SYK ] Similarity = 0.8095238095238095  
[ AZN , ADBE ] Similarity = 0.8095238095238095  
[ BDX , MMM ] Similarity = 0.9523809523809523  
[ BDX , ADP ] Similarity = 0.8571428571428571  
[ BDX , IBM ] Similarity = 0.8095238095238095  
[ BDX , INTC ] Similarity = 0.9523809523809523  
[ BDX , ITW ] Similarity = 0.8571428571428571  
[ BDX , TXN ] Similarity = 0.9047619047619048  
[ BMY , LLY ] Similarity = 0.8095238095238095  
[ BMY , NVS ] Similarity = 0.8571428571428571  
[ BMY , SYK ] Similarity = 0.9047619047619048  
[ BMY , ADBE ] Similarity = 0.9047619047619048  
[ CVS , UNH ] Similarity = 0.9047619047619048  
[ JNJ , CSCO ] Similarity = 0.8571428571428571  
[ JNJ , INTC ] Similarity = 0.8095238095238095  
[ JNJ , SAP ] Similarity = 0.8095238095238095

[ LLY , SYK ] Similarity = 0.8095238095238095  
[ LLY , ADBE ] Similarity = 0.8095238095238095  
[ LLY , SAP ] Similarity = 0.8095238095238095  
[ MDT , PFE ] Similarity = 0.8095238095238095  
[ MDT , FB ] Similarity = 0.8095238095238095  
[ MDT , GOOG ] Similarity = 0.8095238095238095  
[ MDT , INTU ] Similarity = 0.8571428571428571  
[ MDT , MSFT ] Similarity = 0.9047619047619048  
[ MDT , ORCL ] Similarity = 0.8571428571428571  
[ MMM , ADP ] Similarity = 0.9047619047619048  
[ MMM , IBM ] Similarity = 0.8095238095238095  
[ MMM , INTC ] Similarity = 0.9047619047619048  
[ MMM , ITW ] Similarity = 0.9047619047619048  
[ MMM , TXN ] Similarity = 0.8571428571428571  
[ MRK , NVS ] Similarity = 0.8095238095238095  
[ MRK , SYK ] Similarity = 0.8571428571428571  
[ MRK , ADBE ] Similarity = 0.8571428571428571  
[ MRK , SAP ] Similarity = 0.8571428571428571  
[ MRK , VMW ] Similarity = 0.8095238095238095  
[ PFE , MSFT ] Similarity = 0.8095238095238095  
[ SYK , ADBE ] Similarity = 1.0  
[ SYK , CRM ] Similarity = 0.8095238095238095  
[ AAPL , AVGO ] Similarity = 0.8095238095238095  
[ AAPL , IBM ] Similarity = 0.8571428571428571  
[ AAPL , INTC ] Similarity = 0.8095238095238095  
[ AAPL , TSM ] Similarity = 0.8095238095238095  
[ ADBE , CRM ] Similarity = 0.8095238095238095  
[ ADP , IBM ] Similarity = 0.8095238095238095  
[ ADP , INTC ] Similarity = 0.8095238095238095  
[ ADP , ITW ] Similarity = 0.9047619047619048  
[ CSCO , VMW ] Similarity = 0.8095238095238095  
[ FB , GOOG ] Similarity = 0.9047619047619048  
[ FB , ORCL ] Similarity = 0.9523809523809523  
[ GOOG , MSFT ] Similarity = 0.8095238095238095  
[ GOOG , ORCL ] Similarity = 0.9523809523809523  
[ GOOG , QCOM ] Similarity = 0.8571428571428571  
[ IBM , INTC ] Similarity = 0.8571428571428571  
[ IBM , ITW ] Similarity = 0.8095238095238095  
[ IBM , TXN ] Similarity = 0.8095238095238095  
[ INTC , ITW ] Similarity = 0.8095238095238095  
[ INTC , TXN ] Similarity = 0.9523809523809523  
[ INTU , MSFT ] Similarity = 0.9523809523809523  
[ INTU , SAP ] Similarity = 0.8095238095238095  
[ INTU , TSM ] Similarity = 0.8095238095238095

[ MSFT , VMW ] Similarity = 0.8095238095238095  
[ ORCL , QCOM ] Similarity = 0.8095238095238095  
[ TSM , TXN ] Similarity = 0.8095238095238095

### Python program:

```
import pandas as pd
import numpy as np
import operator
output_address = "C:\\Users\\HP\\Desktop\\HW4\\output.txt"

def OutputTXT(file_address,string):
    with open(file_address,'w') as f:
        f.write(string)

def Outputdf(df):
    df.to_csv('C:\\Users\\HP\\Desktop\\HW4\\output.csv')

def OpenCSV(tickers,address):
    dict={}
    for ticker in tickers:
        dict[ticker]=pd.read_csv(address+"\\\\"+str(ticker)+".csv",index_col=0)
    return dict

consumer_tickers=['AMT', 'AMZN', 'BAM', 'CCI', 'CMCSA', 'COST', 'DIS', 'HD', 'LOW', 'LVS',
'MAR', 'MCD', 'NFLX', 'PLD', 'RELX', 'SBUX', 'SPG', 'TGT', 'TJX', 'WMT']
finance_tickers=['AXP', 'BAC', 'BK', 'BLK', 'C', 'CME', 'COF', 'GS', 'HDB', 'ITUB', 'JPM', 'LFC', 'MS',
'PNC', 'RY', 'SAN', 'SPGI', 'TD', 'USB', 'WFC']
healthcare_tickers=['ABBV', 'ABT', 'AMGN', 'ANTM', 'AZN', 'BDX', 'BMY', 'CVS', 'GILD', 'GSK',
'JNJ', 'LLY', 'MDT', 'MMM', 'MRK', 'NVO', 'NVS', 'PFE', 'SYK', 'UNH']
technology_tickers=['AAPL', 'ADBE', 'ADP', 'AVGO', 'CRM', 'CSCO', 'FB', 'GOOG', 'IBM', 'INTC',
'INTU', 'ITW', 'MSFT', 'NVDA', 'ORCL', 'QCOM', 'SAP', 'TSM', 'TXN', 'VMW']
consumer_dict=OpenCSV(consumer_tickers,'C:\\Users\\HP\\Desktop\\HW4\\Consumer')
finance_dict=OpenCSV(finance_tickers,'C:\\Users\\HP\\Desktop\\HW4\\Finance')
healthcare_dict=OpenCSV(healthcare_tickers,'C:\\Users\\HP\\Desktop\\HW4\\Healthcare')
technology_dict=OpenCSV(technology_tickers,'C:\\Users\\HP\\Desktop\\HW4\\Technology')
columns=['Gross Margin', 'Operating Margin', 'Payout Ratio', 'Tax Rate',
'Net Margin', 'Asset Turnover', 'Return on Assets',
'Financial Leverage', 'Return on Equity', 'Return on Invested Capital',
'Interest Coverage', 'Current Ratio', 'Quick Ratio',
'Financial Leverage', 'Debt to Equity Ratio', 'Fixed Assets Turnover',
'Asset Turnover']
tickers=consumer_tickers+finance_tickers+healthcare_tickers+technology_tickers

dict={**consumer_dict,**finance_dict,**healthcare_dict,**technology_dict}

def getData(ticker, date, type):
    return float(dict[ticker].loc[date,type])
```

```

def getDict(date, type):
    new_dict={}
    for ticker in tickers:
        new_dict[ticker]=getData(ticker,date,type)
    return new_dict

```

```

def getDataframe(date):
    list1=[]
    for ticker in tickers:
        list2=[]
        for type in columns:
            list2.append(getData(ticker,date,type))
        list1.append(list2)
    df=pd.DataFrame(list1,index=tickers,columns=columns)
    return df

```

```

#normalize data
def ScaleDataframe(df):
    scaled_df=(df-df.min())/(df.max()-df.min())
    return scaled_df

```

```

def RandomArray(dimension):
    return np.random.random(dimension)

```

```

class Centroid:      #a single centroid point
    def __init__(self,position):
        self.position=position
        self.ticker_list=[]
        self.point_list=[]      #Assign different points to point_list for each centroid
        self.previous_ticker_list=[]      #Determine when the algorithm is finished.

```

```

class Kmeans:
    def __init__(self, k,df, maximum_iters):      #k: k value,
        self.k = k
        self.centroid_list = []      #list of centroids
        self.df = df
        self.data=df.values
        self.dimension = df.values.shape[1]      #df.values.shape[1] is dimension of centroid
coordinate.
        self.maximum_iters=maximum_iters
        self.count = 0

```

```

#initialize random centroid_list
for _ in range(k):
    position_array=np.array(RandomArray(self.dimension))
    self.centroid_list.append(Centroid(position_array))

def assign_centroid(self,x):
    """
    return centroid closest to a certain point
    """

    distances = {}
    for centroid in self.centroid_list:
        distances[centroid] = np.linalg.norm(centroid.position - x) #This is a
dictionary, key is centroid, value is distance from data to centroid
    closest_centroid = min(distances.items(), key = operator.itemgetter(1))[0] #return
centroid which has min of distance
    return closest_centroid

def fit(self):
    """
    Fit point_list in self.data
    Assign the point_list the centroid
    Update centroid location based on assigned point_list
    """

    terminate = 0
    while self.count < self.maximum_iters and terminate < self.k:
        terminate = 0
        #Erase last allocation result.
        for centroid in self.centroid_list:
            centroid.previous_ticker_list = centroid.ticker_list
            centroid.point_list=[]
            centroid.ticker_list=[]

        #add points to different centroids.
        for i, point in enumerate(self.data):
            closest_centroid = self.assign_centroid(point)
            closest_centroid.point_list.append(point)
            closest_centroid.ticker_list.append(self.df.index[i])

        #determine terminate condition
        for i in range(self.k):
            if (self.centroid_list[i].previous_ticker_list == self.centroid_list[i].ticker_list
and len(self.centroid_list[i].ticker_list)>0):
                terminate += 1

```



```

        for i in range(self.k):
            array = np.array([0.0]*self.dimension)
            if len(self.centroid_list[i].point_list)>0:
                for j in range(len(self.centroid_list[i].point_list)):
                    array += self.centroid_list[i].point_list[j]
                array = array/(len(self.centroid_list[i].point_list))
                self.centroid_list[i].position = array
            elif len(self.centroid_list[i].point_list)==0:
                for j in range(self.df.values.shape[0]):
                    array += self.data[j]
                array = array/self.df.values.shape[0]
                self.centroid_list[i].position = array
        self.count += 1
    def __str__(self):
        string=""
        string+="Iteration count ="+str(self.count)+"\n"
        for i in range(self.k):
            string+="Group   "+str(i)+"   with   "+str(len(self.centroid_list[i].ticker_list))+
element"+ "\n"
            string+="Ticker list : "+str(self.centroid_list[i].ticker_list)+"\n\n"
        return string
    def __repr__(self):
        string=""
        string+="Iteration count ="+str(self.count)+"\n"
        for i in range(self.k):
            string+="Group   "+str(i)+"   with   "+str(len(self.centroid_list[i].ticker_list))+
element"+ "\n"
            string+="Ticker list : "+str(self.centroid_list[i].ticker_list)+"\n\n"
        return string

def inSameGroup(self,ticker1,ticker2):
    for i in range(self.k):
        if ticker1 in self.centroid_list[i].ticker_list:
            if ticker2 in self.centroid_list[i].ticker_list:
                return True
            else:
                return False

def start(year,k_value,max_iter):
    df=getDataframe(year)
    df.drop('Financial Leverage',axis =1, inplace=True)
    df=ScaleDataframe(df)

```

```

a=Kmeans(k_value,df,max_iter)
a.fit()
string=""
print("Year = ",year, ", K = ",k_value)
print(a)
string+="Year = "+str(year)+"", K = "+str(k_value)+"\n"
string+=str(a)
return string, a

```

```

#main program
string=""
kmeans_list=[]
for year in range(2013,2016):
    for k_value in range(2,9):
        s, kmeans_result= start(year,k_value,100)
        string+=s
        kmeans_list.append(kmeans_result)
string+="*****\n"

```

OutputTXT(output\_address,string)

```

#Similarity
for i, ticker1 in enumerate(tickers):
    for j, ticker2 in enumerate(tickers):
        if i<j:
            count = 0
            threshold = 0.8*len(kmeans_list)
            for a in kmeans_list:
                if a.inSameGroup(ticker1, ticker2):
                    count+=1
            if count > threshold:
                print("[",ticker1,",",ticker2,"] Similarity = ",count/len(kmeans_list))

```

Reference:

<https://www.morningstar.com/>