

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỆ THỐNG THÔNG TIN



BÁO CÁO ĐỒ ÁN
HỆ THỐNG THÔNG TIN KẾ TOÁN

ĐỀ TÀI:
TÍNH TOÁN, TRỰC QUAN HÓA CANSLIM &
4M

Giảng viên hướng dẫn: ThS. Đỗ Duy Thanh

Sinh viên thực hiện:

Huỳnh Quốc Khánh – 19521677

Bùi Sĩ Khoa – 20521461

Doãn Yến Nhung – 20521723

TP. Hồ Chí Minh, tháng 1 năm 2024

LỜI CẢM ƠN

Nhóm xin gửi lời cảm ơn và lòng biết ơn sâu sắc tới ThS. Đỗ Duy Thanh (giảng viên môn Hệ thống thông tin kế toán). Thầy đã tận tình hướng dẫn, giúp đỡ, cung cấp tài liệu và tạo điều kiện cho nhóm tác giả hoàn thành đồ án này.

Trong thời gian thực hiện đề tài, nhóm đã cố gắng vận dụng những kiến thức nền tảng đã tích lũy đồng thời kết hợp với việc học hỏi và nghiên cứu những kiến thức mới để ứng dụng vào thực hiện đề tài này. Mặc dù đã hết sức cố gắng, nhưng chắc chắn đồ án vẫn còn những thiếu sót. Chính vì vậy, nhóm rất mong nhận được những sự góp ý quý báu từ thầy nhằm hoàn thiện những kiến thức còn thiếu và là hành trang để nhóm thực hiện tiếp các đề tài khác trong tương lai.

Xin chân thành cảm ơn Thầy!

Thành phố Hồ Chí Minh, tháng 01 năm 2024

Nhóm sinh viên thực hiện

[illegible]

MỤC LỤC

Chương 1	GIỚI THIỆU ĐỒ ÁN	6
1.1	Lý do chọn đề tài	6
1.2	Tổng quan dữ liệu.....	6
1.3	Công cụ và thư viện	6
Chương 2	PHƯƠNG PHÁP NGHIÊN CỨU	7
2.1	Canslim là gì ?	7
2.1.1	Định nghĩa	7
2.1.2	Ba nguyên tắc của Canslim.....	7
2.1.3	Phương pháp tính điểm Canslim.....	8
2.2	4M là gì ?	10
2.2.1	Định nghĩa	10
2.2.2	Ưu và nhược điểm.....	11
2.2.3	Phương pháp tính điểm 4M	11
Chương 3	PHÂN TÍCH VÀ ĐƯA RA KẾT QUẢ.....	13
3.1	Tính Canslim	16
3.2	Tính 4M	20
3.3	Khởi tạo server local hiển thị Dashboard	25
Chương 4	KẾT LUẬN.....	28
4.1	Ưu điểm.....	28
4.2	Nhược điểm.....	28
4.3	Hướng phát triển.....	28

Chương 1 GIỚI THIỆU ĐỒ ÁN

1.1 Lý do chọn đề tài

Hiện nay, mỗi nhà đầu tư sẽ chọn cho mình phương pháp đầu tư phù hợp với cá tính và kỳ vọng của bản thân. Sự quan trọng của việc kết hợp giữa phân tích cơ bản với phân tích kỹ thuật trong nghiên cứu, đầu tư cổ phiếu và ví von sự kết hợp này như nội công cùng chiêu thức, một phần quan trọng tạo nên tinh hoa võ thuật của người xưa, là nền tảng của Kungfu chứng khoán ngày nay, và bây giờ là một sự kết hợp khác giữa phương pháp đầu tư 4M và Canslim. Chính vì thế nhóm chúng em quyết định lựa chọn đề tài này để có thể tìm hiểu, học hỏi cũng như làm rõ hơn về 2 phương pháp đầu tư đã nêu trên, từ đó đưa ra nhiều đúc kết kinh nghiệm cho bản thân cũng như nắm bắt thêm nhiều kiến thức không chỉ gói gọn trong mỗi lý thuyết đã học trên lớp mà còn học hỏi thêm nhiều giá trị thông qua thực tiễn vốn có.

1.2 Tổng quan dữ liệu

Dữ liệu về tài chính của các doanh nghiệp được crawled về và xử lý bởi nhóm

Nguồn dữ liệu: <https://finance.vietstock.vn/>

1.3 Công cụ và thư viện

- Ngôn ngữ lập trình: Python.
- Thư viện:

```
import pandas as pd
import numpy as np
import panel as pn
import numpy_financial as npf
import matplotlib.pyplot as plt
import pandas as pd
from math import pi
from matplotlib.figure import Figure
from matplotlib import cm
from matplotlib.backends.backend_agg import FigureCanvas
pn.extension('tabulator')
import hvplot.pandas
```

Chương 2 PHƯƠNG PHÁP NGHIÊN CỨU

2.1 Canslim là gì ?

2.1.1 Định nghĩa

- **Canslim** là một phương pháp chọn mua chứng khoán nổi tiếng được nhiều người biết đến do William O'Neil - người sáng lập ra tờ Investor's Business Daily, sáng tạo ra. Phương pháp này chọn mua chứng khoán bao hàm cả phân tích cơ bản và phân tích kỹ thuật.
- Mỗi ký tự trong từ **Canslim** đại diện cho một trong 7 đặc điểm chính của những cổ phiếu thành công trong quá khứ.
- C = Lãi ròng trên mỗi cổ phần quý hiện tại hay gọi nôm na là “EPS quý hiện tại” (Current Earning Per Share), tiêu chuẩn này yêu cầu EPS tăng dần và tăng càng cao càng tốt
- A = Tỷ lệ tăng trưởng lợi nhuận hàng năm (Annual Earning Increases). Điều này có nghĩa là cổ phiếu tốt là cổ phiếu có mức gia tăng lợi nhuận đều đặn năm sau so với năm trước
- N = Những thông tin mới về công ty như sản phẩm mới, ban lãnh đạo mới, định giá mới...
- S = Quan hệ cung cầu của cổ phiếu (Supply and Demand). Cổ phiếu cũng là một loại hàng hoá, do vậy, giá cả chịu sự điều chỉnh của quan hệ cung cầu.
- L = Xem xét vai trò của cổ phiếu đó trên thị trường là cổ phiếu “dẫn đầu” hay chỉ là cổ phiếu “đội sổ” (leader/laggard).
- I = Sự quan tâm của các tổ chức, định chế tài chính lớn đến cổ phiếu (Institutional Sponsorship).
- M = Xu hướng thị trường (Market Direction) là yếu tố quan trọng quyết định sự thành bại của phương án đầu tư.

2.1.2 Ba nguyên tắc của Canslim

- **Nguyên tắc 1:** Chỉ mua cổ phiếu khi thị trường tăng giá; thực hiện các biện pháp phòng thủ khi những “cú tát của con gấu” bắt đầu xuất hiện trên thị trường.
- **Nguyên tắc 2:** Tập trung vào các công ty có tăng trưởng lợi nhuận cao, sản phẩm hoặc dịch vụ mới, mang tính cách mạng.

- **Nguyên tắc 3:** Mua các cổ phiếu được nhiều nhà đầu tư tổ chức thu gom; tránh xa các cổ phiếu khi các nhà đầu tư tổ chức bán tháo nó.

2.1.3 Phương pháp tính điểm Canslim

Bước 1: Dựa vào tiêu chí EPS và Sale ta lập được một bảng như trên

VNM									
Tiêu chí SALE	1.Quý gần nhất (C)				Q2 2022	Q2 2023			
					14,930,275	15,194,825			
	1.Quý trước đó gần nhất (C)				Q1 2022	Q1 2023			
					13,877,826	13,918,392			
	Trailing 12 tháng gần nhất (A)	Q3 2021	Q4 2021	Q1 2022	Q2 2022	Q3 2022	Q4 2022	Q1 2023	Q2 2023
	Trailing 12 tháng gần nhất trước đó (A)	Q2 2021	Q3 2021	Q4 2021	Q1 2022	Q2 2022	Q3 2022	Q4 2022	Q1 2023
		15,715,806	16,194,052	15,819,037	13,877,826	14,930,275	16,079,491	15,068,655	13,918,392
Tiêu chí EPS	1.Quý gần nhất (C)				Q1 2022	Q1 2023			
					4,891	3,879			
	1.Quý trước đó gần nhất (C)				Q4 2021	Q4 2022			
					5,040	4,075			
	Trailing 12 tháng gần nhất (A)	Q2 2021	Q3 2021	Q4 2021	Q1 2022	Q2 2022	Q3 2022	Q4 2022	Q1 2023
	Trailing 12 tháng gần nhất trước đó (A)	Q1 2021	Q2 2021	Q3 2021	Q4 2021	Q1 2022	Q2 2022	Q3 2022	Q4 2022
		5,752	5,382	5,075	5,040	4,891	4,531	4,231	4,075

Hình 2.1.1: Lập bảng dựa trên EPS và Sale

Bước 2: Lập cột tỉ lệ tăng trưởng dựa vào tổng giá trị của cột phải trừ tổng giá trị của cột trái sau đó chia tổng giá trị của cột trái

Tỷ lệ tăng trưởng
1.77%
0.29%
-0.92%
-2.61%
-20.69%
-19.15%
-18.01%
-16.57%

Hình 2.1.2: Lập cột tỉ lệ tăng trưởng

Bước 3: Lập cột tham chiếu và tỉ trọng từng thành phần

Tham chiếu	Tỷ trọng từng thành phần
25%	20%
25%	15%
20%	15%
20%	10%
20%	15%
20%	10%
20%	10%
20%	5%

Hình 2.1.3: Lập cột tham chiếu và tỉ trọng từng thành phần

Bước 4: Lập cột C, A theo format $=IF(L6<0,0,IF(L6>M6,N6*100,(L6/M6)*N6*100))$ và tính ra Tổng điểm bằng tổng của cột C, A

Hình 2.1.4: Lập cột C, A và Tổng điểm

2.2 4M là gì ?

2.2.1 Định nghĩa

- 4M là từ viết tắt của 4 tiêu chí lựa chọn cổ phiếu theo quan điểm của Phil Town, bao gồm:

Meaning: Đầu tư vào những doanh nghiệp dễ hiểu

Moat: Đầu tư vào những doanh nghiệp có lợi thế cạnh tranh lớn

Management: Đầu tư vào những doanh nghiệp có ban lãnh đạo tốt

Margin of safety: Nên đầu tư vào những doanh nghiệp có biên an toàn

- Theo ông, những cổ phiếu thỏa mãn cả 4 tiêu chí này sẽ là khoản đầu tư tuyệt vời và tích lũy chúng là cách để những nhà đầu tư cá nhân trở nên giàu có.

2.2.2 Ưu và nhược điểm

*Ưu điểm:

- Dễ hiểu, dễ hình dung với những nhà đầu tư nhỏ lẻ
- Dễ thực hiện ở các chỉ tiêu định tính

*Nhược điểm:

- Các chỉ tiêu định lượng quá đơn giản

2.2.3 Phương pháp tính điểm 4M

Bước 1: Dựa vào bảng phân tích báo cáo của doanh nghiệp theo năm ta lập được một bảng như hình

	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	Tham chiếu	tỷ trọng
Chỉ số														
Sales	26,561,574	30,948,602	34,976,928	40,080,385	46,794,339	51,041,076	52,561,950	56,318,123	59,636,286	60,919,165	59,956,247		20%	15%
EPS	10,471	7,999	6,870	7,296	7,298	7,094	6,645	6,076	6,112	5,040	4,075		20%	20%
BVPS	27,881	21,051	19,678	17,435	15,437	16,450	15,086	17,073	16,102	17,154	15,702		15%	5%
OPC LCDTKD = Lưu chuyển Tiền thuần từ HĐKD	6,629,825	7,711,678	7,308,841	9,271,226	11,160,291	12,226,418	11,876,513	12,797,090	13,539,381	12,727,620	10,491,065		15%	15%
Nợ dài hạn năm gần nhất													<3*LN	10%
Effectiveness (Doanh thu/Tài sản)	1.35	1.35	1.36	1.46	1.59	1.47	1.41	1.26	1.23	1.14	1.24		10%	
Efficiency (Lợi nhuận / Doanh thu)	21.91	21.11	17.35	19.38	20.01	20.14	19.42	18.74	18.84	17.45	14.31		10%	
Productivity (OPC LCDTKD/ Lợi nhuận)													10%	5%
ROA	32.99	30.70	24.95	29.20	32.89	32.15	28.40	25.79	23.83	20.70	16.73		15%	10%
ROE	41.61	39.55	32.61	38.29	43.16	44.49	40.79	37.79	35.02	30.31	24.80		20%	5%
ROIC													15%	15%

Hình 2.2.1: Lập bảng dựa trên báo cáo năm

Bước 2: Tính chỉ số \ Trọng số quá khứ 1, 3, 5, 10 năm của doanh nghiệp như bảng ở dưới với hàm Rate theo format =**RATE(\$B\$2,-'Data4M-BSR'!F3,'Data4M-BSR'!G3)**

	1	3	5	10	Tham chiếu	Tỷ trọng	Điểm tp
Chỉ số/Trọng số quá khứ(đỏ)	30%	40%	30%				
Sales Growth Rate	-2%	2%	3%		20%	15%	6.75
EPS Growth Rate	-19%		-12%	-10%	20%	20%	-69.40
BVPS Growth Rates	-8%		-3%	-1%	15%	5%	-26.12
Tăng trưởng OPC LCDTKD			-6%		15%	15%	-17.09
Nợ dài hạn năm gần nhất		Lợi nhuận năm gần nhất			<3*LN	10%	#VALUE!
Effectiveness(Doanh thu/Tài sản)	8%		-0.62%	-3%	10%	5%	12.03
Efficiency(Lợi nhuận/Doanh thu)	-18%		-8.60%	-7%	10%		-108.19
Productivity(OPC LCDTKD/Lợi nhuận)					10%		0.00
ROA	-19%		-13%	-12%	15%	10%	-98.67
ROE	-18%		-13%	-11%	20%	5%	-70.01
ROIC					15%	15%	0.00

Hình 2.2.2: Tính chỉ số \ Trọng số quá khứ

Bước 3: Lập cột tham chiếu và tỷ trọng

Tham chiếu	tỷ trọng
20%	15%
20%	20%
15%	5%
15%	15%
<3*LN	10%
10%	
10%	
10%	5%
15%	10%
20%	5%
15%	15%

Hình 2.2.3: Lập cột tham chiếu và tỷ trọng

Bước 4: Tính điểm thành phần dựa trên format

=SUM(IF(B4>=F4,\$B\$3,(B4/F4)*\$B\$3),IF(C4>=F4,\$C\$3,(C4/F4)*\$C\$3),IF(D4>=F4,\$D\$3,(D4/F4)*\$D\$3))*100*G4 rồi lấy tổng cột điểm thành phần

Điểm tp	Tổng
6.75	
-69.40	
-26.12	
-17.09	
#VALUE!	
12.03	
-108.19	
0.00	
-98.67	
-70.01	
0.00	-370.70

Hình 2.2.4: Lập cột điểm thành phần và tổng

Chương 3 PHÂN TÍCH VÀ ĐƯA RA KẾT QUẢ

Bước 1: Tiến crawl dữ liệu và chuyển dữ liệu đã crawl sang file csv

```
import pandas as pd
from selenium.webdriver.common.by import By
from time import sleep

browser.get("https://finance.vietstock.vn/VNM/tai-chinh.htm?tab=BCTT")
sleep(5)

[ ] ChonKy_1= browser.find_element("xpath","//*[@id="finance-content"]/div/div/div[2]/div/div[2]/div[1]/div[1]/select[1]/option[8]')
ChonKy_1.click()

[ ] ChonNam_1= browser.find_element("xpath","//*[@id="finance-content"]/div/div/div[2]/div/div[2]/div[1]/div[1]/select[2]/option[1]')
ChonNam_1.click()

[ ] ChonKy_1= browser.find_element("xpath","//*[@id="finance-content"]/div/div/div[2]/div/div[2]/div[1]/div[1]/select[1]/option[6]')
ChonKy_1.click()

[ ] CT KD= browser.find_element("xpath","//*[@id="tbl-data-BCTT-KQ"]')

table_CTKD = []

table_header_CTKD = CT KD.find_elements(By.XPATH,"thead/tr/th")

header_row_CTKD = []
for header in table_header_CTKD:
    header_row_CTKD.append(header.text)

table_data_CTKD = CT KD.find_elements(By.XPATH,"tbody/tr")
index_CTKD=0
for row in table_data_CTKD:
```

	2016	2017	2018	2019	2020	2021	2022
Giai đoạn	01/01-31/12	01/01-31/12	01/01-31/12	01/01-31/12	01/01-31/12	01/01-31/12	01/01-31/12
Hợp nhất	Hợp nhất	Hợp nhất	Hợp nhất	Hợp nhất	Hợp nhất	Hợp nhất	Hợp nhất
Kiểm toán	Kiểm toán	Kiểm toán	Kiểm toán	Kiểm toán	Kiểm toán	Kiểm toán	Kiểm toán
Công ty kiểm toán	KPMG	KPMG	KPMG	KPMG	KPMG	KPMG	KPMG
Ý kiến kiểm toán	Chấp nhận toàn phần	Chấp nhận toàn phần	Chấp nhận toàn phần	Chấp nhận toàn phần	Chấp nhận toàn phần	Chấp nhận toàn phần	Chấp nhận toàn phần
Doanh thu thuần về bán hàng và cung cấp dịch vụ	46,794	51,041	52,562	56,318	59,636	60,919	59,956
Giá vốn hàng bán	24,459	26,807	27,951	29,746	31,968	34,641	36,059
Lợi nhuận gộp về bán hàng và cung cấp dịch vụ	22,336	24,234	24,611	26,572	27,669	26,278	23,897
Doanh thu hoạt động tài chính	723	816	760	807	1,581	1,215	1,380
Chi phí tài chính	102	87	118	187	309	202	618
Chi phí bán hàng	10,759	11,537	12,266	12,993	13,447	12,951	12,548
Chi phí quản lý doanh nghiệp	1,053	1,268	1,133	1,396	1,958	1,567	1,596
Lợi nhuận thuần từ hoạt động kinh doanh	11,160	12,226	11,877	12,797	13,539	12,728	10,491
Lợi nhuận khác	77	3	175	-1	-21	195	4
Phần lợi nhuận/lỗ từ công ty liên kết liên doanh							
Tổng lợi nhuận kế toán trước thuế	11,238	12,229	12,052	12,796	13,519	12,922	10,496
Lợi nhuận sau thuế thu nhập doanh nghiệp	9,364	10,278	10,206	10,554	11,236	10,633	8,578
Lợi nhuận sau thuế của cổ đông Công ty mẹ	9,350	10,296	10,227	10,581	11,099	10,532	8,516
Lãi cơ bản trên cổ phiếu (VNĐ)	5,831	6,355	5,295	5,478	4,770	4,517	3,632

```

CDKT= browser.find_element("xpath", '//*[@id="tbl-data-BCTT-CD"]')
# CDKT= browser.find_element("xpath", '//*[@id="tbl-data-BCTT-CD"]')

table_CDKT = []

table_header_CDKT = CDKT.find_elements(By.XPATH,"thead/tr/th")

header_row_CDKT = []
for header in table_header_CDKT:
    header_row_CDKT.append(header.text)

table_data_CDKT = CDKT.find_elements(By.XPATH,"tbody/tr")
index_CDKT=[]
for row in table_data_CDKT:
    columns = row.find_elements(By.XPATH,"./td")
    table_row = []
    count=0
    for column in columns:
        if(count==0):
            index_CDKT.append(column.text)
        else:
            table_row.append(column.text)
            count+=1
    table_CDKT.append(table_row)

header_row_CDKT.pop(0)
df_CDKT = pd.DataFrame(table_CDKT, index = index_CDKT,columns =header_row_CDKT)
df_CDKT

```

	2016	2017	2018	2019	2020	2021	2022
Giai đoạn	01/01-31/12	01/01-31/12	01/01-31/12	01/01-31/12	01/01-31/12	01/01-31/12	01/01-31/12
Hợp nhất	Hợp nhất	Hợp nhất	Hợp nhất	Hợp nhất	Hợp nhất	Hợp nhất	Hợp nhất
Kiểm toán	Kiểm toán	Kiểm toán	Kiểm toán	Kiểm toán	Kiểm toán	Kiểm toán	Kiểm toán
Công ty kiểm toán	KPMG	KPMG	KPMG	KPMG	KPMG	KPMG	KPMG
Ý kiến kiểm toán	Chấp nhận toàn phần	Chấp nhận toàn phần	Chấp nhận toàn phần	Chấp nhận toàn phần	Chấp nhận toàn phần	Chấp nhận toàn phần	Chấp nhận toàn phần
Tài sản ngắn hạn	18,674	20,307	20,560	24,722	29,666	36,110	31,560
Tiền và các khoản tương đương tiền	655	963	1,523	2,665	2,111	2,349	2,300
Các khoản đầu tư tài chính ngắn hạn	10,454	10,562	8,674	12,436	17,314	21,026	17,414
Các khoản phải thu ngắn hạn	2,867	4,592	4,639	4,503	5,187	5,822	6,100
Hàng tồn kho	4,522	4,021	5,526	4,983	4,905	6,773	5,538
Tài sản ngắn hạn khác	176	170	198	134	148	141	208
Tài sản dài hạn	10,705	14,360	16,806	19,978	18,767	17,222	16,922
Tài sản cố định	8,321	10,609	13,365	14,894	13,854	12,707	11,903
Bất động sản đầu tư	137	95	90	62	60	60	58
Các khoản đầu tư tài chính dài hạn	614	555	1,069	987	973	744	743
Tổng cộng tài sản	29,379	34,667	37,366	44,700	48,432	53,332	48,483
Nợ phải trả	6,973	10,794	11,095	14,969	14,785	17,482	15,666
Nợ ngắn hạn	6,457	10,196	10,640	14,443	14,213	17,068	15,308
Nợ dài hạn	515	599	455	526	573	414	358
Vốn chủ sở hữu	22,406	23,873	26,271	29,731	33,647	35,850	32,817
Vốn đầu tư của chủ sở hữu	14,515	14,515	17,417	17,417	20,900	20,900	20,900
Thặng dư vốn cổ phần	261	261				34	34
Lợi nhuận sau thuế chưa phân phối	5,592	5,737	7,155	7,875	6,910	7,594	3,353
Lợi ích của cổ đông thiểu số							
Tổng cộng nguồn vốn	29,379	34,667	37,366	44,700	48,432	53,332	48,483

```
[ ] df_CTKD.to_csv('CTKD_quy.csv',sep='\t',encoding='utf-16')
    df_CDKT.to_csv('CDKT_quy.csv',sep='\t',encoding='utf-16')
    df_CSTC.to_csv('CSTC_quy.csv',sep='\t',encoding='utf-16')

[ ] df_CTKD.to_csv('CTKD_nam.csv',sep='\t',encoding='utf-16')
    df_CDKT.to_csv('CDKT_nam.csv',sep='\t',encoding='utf-16')
    df_CSTC.to_csv('CSTC_nam.csv',sep='\t',encoding='utf-16')
```

Hình 3.1: Crawl dữ liệu từ website

Bước 2: Tiến hành import các thư viện cần thiết

▼ Import các thư viện cần thiết

```
[1]: import pandas as pd
import numpy as np
import panel as pn
import numpy_financial as npf
import matplotlib.pyplot as plt
import pandas as pd
from math import pi

from matplotlib.figure import Figure
from matplotlib import cm
from matplotlib.backends.backend_agg import FigureCanvas
pn.extension('tabulator')

import hvplot.pandas
```

Hình 3.2: Import thư viện cần thiết

Bước 3: Nhập mã doanh nghiệp

▼ Chọn doanh nghiệp từ mã

```
[2]: business_code = input("Nhập mã doanh nghiệp: ")
      Nhập mã doanh nghiệp: BVH
```

Hình 3.3: Nhập mã doanh nghiệp

3.1 Tính Canslim

Bước 1: Import dữ liệu doanh nghiệp

1. Import dữ liệu doanh nghiệp

```
: df_CTKD = pd.read_csv("Data/"+business_code+"/CTKD_quy.csv", encoding='utf-16', sep='\t')
df_CDKT = pd.read_csv("Data/"+business_code+"/CDKT_quy.csv", encoding='utf-16', sep='\t')
df_CSTC = pd.read_csv("Data/"+business_code+"/CSTC_quy.csv", encoding='utf-16', sep='\t')
```

Hình 3.1.1: Import dữ liệu doanh nghiệp

Bước 2: Tiến hành tiền xử lý dữ liệu

2. Tiền xử lý dữ liệu

```
] : df_CTKD.drop('Unnamed: 1', axis=1, inplace=True)
df_CTKD.drop('Unnamed: 2', axis=1, inplace=True)
df_CTKD.drop('Unnamed: 3', axis=1, inplace=True)

df_CDKT.drop('Unnamed: 1', axis=1, inplace=True)
df_CDKT.drop('Unnamed: 2', axis=1, inplace=True)
df_CDKT.drop('Unnamed: 3', axis=1, inplace=True)

df_CSTC.drop('Unnamed: 1', axis=1, inplace=True)
df_CSTC.drop('Unnamed: 2', axis=1, inplace=True)
df_CSTC.drop('Unnamed: 3', axis=1, inplace=True)

df_CTKD = df_CTKD.fillna(0)
df_CDKT = df_CDKT.fillna(0)
df_CSTC = df_CSTC.fillna(0)
```

Hình 3.1.2: Tiền xử lý dữ liệu

Bước 3: Khởi tạo mảng dữ liệu

3. Khởi tạo các mảng dữ liệu

```
] : col_canslim = df_CTKD.columns
ti_le_tang_truong = []
c=[]
a=[]
```

Hình 3.1.3: Khởi tạo mảng dữ liệu

Bước 4: Tiêu chí sale


```

sales = df_CTKD.loc[6] #Doanh thu thuần hoạt động kinh doanh bảo hiểm
sales_dict = list(sales)
sales_dict.remove('Doanh thu thuần hoạt động kinh doanh bảo hiểm')
sales = []
for i in range(0, len(sales_dict)):
    sales.append(float(sales_dict[i].replace(',', '.')))

# I Quý gần nhất (C)
one_quy_gan_nhat_sale = []
one_quy_gan_nhat_sale.append(sales[len(sales)-5])
one_quy_gan_nhat_sale.append(sales[len(sales)-1])
tang_truong_c1 = (one_quy_gan_nhat_sale[len(one_quy_gan_nhat_sale)-1] - one_quy_gan_nhat_sale[len(one_quy_gan_nhat_sale)-2]) / one_quy_gan_nhat_sale[len(one_quy_gan_nhat_sale)-2] * 100
ti_le_tang_truong.append(tang_truong_c1)
if(tang_truong_c1 < 0):
    c.append(0)
elif(tang_truong_c1 > 25):
    c.append(15)
else:
    c.append((tang_truong_c1/25)*15)

#II Quý trước đó gần nhất (C)
one_quy_truoc_gan_nhat_sale = []
one_quy_truoc_gan_nhat_sale.append(sales[len(sales)-6])
one_quy_truoc_gan_nhat_sale.append(sales[len(sales)-2])
tang_truong_c2 = (one_quy_truoc_gan_nhat_sale[len(one_quy_truoc_gan_nhat_sale)-1] - one_quy_truoc_gan_nhat_sale[len(one_quy_truoc_gan_nhat_sale)-2]) / one_quy_truoc_gan_nhat_sale[len(one_quy_truoc_gan_nhat_sale)-2] * 100
ti_le_tang_truong.append(tang_truong_c2)
if(tang_truong_c2 < 0):
    c.append(0)
elif(tang_truong_c2 > 25):
    c.append(10)
else:
    c.append((tang_truong_c2/25)*10)

#Trailing 12 tháng gần nhất (A)
trailing_gan_sum1_p1 = 0
trailing_gan_sum2_p1 = 0
for i in range(1,5):
    trailing_gan_sum1_p1 += sales[i]
for i in range(5, len(sales)):
    trailing_gan_sum2_p1 += sales[i]
tang_truong_a1 = (trailing_gan_sum2_p1 - trailing_gan_sum1_p1) / trailing_gan_sum1_p1 * 100
ti_le_tang_truong.append(tang_truong_a1)
if(tang_truong_a1 < 0):
    a.append(0)
elif(tang_truong_a1 > 20):
    a.append(10)
else:
    a.append((tang_truong_a1/20)*10)

#Trailing 12 tháng gần nhất trước đó (A)
trailing_truoc_sum1_p2 = 0
trailing_truoc_sum2_p2 = 0
for i in range(0,4):
    trailing_truoc_sum1_p2 += sales[i]
for i in range(4, len(sales)-1):
    trailing_truoc_sum2_p2 += sales[i]
tang_truong_a2 = (trailing_truoc_sum2_p2 - trailing_truoc_sum1_p2) / trailing_truoc_sum1_p2 * 100
ti_le_tang_truong.append(tang_truong_a2)
if(tang_truong_a2 < 0):
    a.append(0)
elif(tang_truong_a2 > 20):
    a.append(5)
else:
    a.append((tang_truong_a2/20)*5)

```

Hình 3.1.4: Tiêu chí sale

Bước 5: Tiêu chí EPS

```

: eps = df_CTKD.loc[16] #'Lãi cơ bản trên cổ phiếu (VND)'
eps_dict= list(eps)
eps_dict.remove('Lãi cơ bản trên cổ phiếu (VND)')
eps = []
for i in range(0,len(eps_dict)):
    eps.append(float(eps_dict[i].replace(',','.')))

#1 Quý gần nhất (C)
one_quy_gan_nhat_eps = []
one_quy_gan_nhat_eps.append(eps[len(eps)-5])
one_quy_gan_nhat_eps.append(eps[len(eps)-1])
tang_truong_c3= (one_quy_gan_nhat_eps[len(one_quy_gan_nhat_eps)-1]-one_quy_gan_nhat_eps[len(one_quy_gan_nhat_eps)-2])/one_quy_gan_nhat_eps[len(one_quy_gan_nhat_eps)-2]*100
ti_le_tang_truong.append(tang_truong_c3)
if(tang_truong_c3 < 0):
    c.append(0)
elif(tang_truong_c3 > 25):
    c.append(20)
else:
    c.append((tang_truong_c3/25)*20)

#1 Quý trước đó gần nhất (C)
one_quy_truoc_gan_nhat_eps = []
one_quy_truoc_gan_nhat_eps.append(eps[len(eps)-6])
one_quy_truoc_gan_nhat_eps.append(eps[len(eps)-2])
tang_truong_c4= (one_quy_truoc_gan_nhat_eps[len(one_quy_truoc_gan_nhat_eps)-1]-one_quy_truoc_gan_nhat_eps[len(one_quy_truoc_gan_nhat_eps)-2])/one_quy_truoc_gan_nhat_eps[len(one_quy_truoc_gan_nhat_eps)-2]*100
ti_le_tang_truong.append(tang_truong_c4)
if(tang_truong_c4 < 0):
    c.append(0)
elif(tang_truong_c4 > 25):
    c.append(15)
else:
    c.append((tang_truong_c4/25)*15)

#Trailing 12 tháng gần nhất (A)
trailing_gan_sum1_p1_eps = 0
trailing_gan_sum2_p1_eps = 0
for i in range(1,5):
    trailing_gan_sum1_p1_eps += eps[i]
for i in range(5,len(eps)):
    trailing_gan_sum2_p1_eps += eps[i]
tang_truong_a3 = (trailing_gan_sum2_p1_eps-trailing_gan_sum1_p1_eps)/trailing_gan_sum1_p1_eps*100
ti_le_tang_truong.append(tang_truong_a3)
if(tang_truong_a3 < 0):
    a.append(0)
elif(tang_truong_a3 > 20):
    a.append(15)
else:
    a.append((tang_truong_a3/20)*15)

#Trailing 12 tháng gần nhất trước đó (A)
trailing_truoc_sum1_p2_eps = 0
trailing_truoc_sum2_p2_eps = 0
for i in range(0,4):
    trailing_truoc_sum1_p2_eps += eps[i]
for i in range(4,len(eps)-1):
    trailing_truoc_sum2_p2_eps += eps[i]
tang_truong_a4 = (trailing_truoc_sum2_p2_eps-trailing_truoc_sum1_p2_eps)/trailing_truoc_sum1_p2_eps*100
ti_le_tang_truong.append(tang_truong_a4)
if(tang_truong_a4 < 0):
    a.append(0)
elif(tang_truong_a4 > 20):
    a.append(10)
else:
    a.append((tang_truong_a4/20)*10)

```

Hình 3.1.5: Tiêu chí EPS

Bước 6: Trực quan hóa kết quả

```

tieu_chi = ['1 Quý gần nhất (C)', '1 Quý trước đó gần nhất (C)', 'Trailing 12 tháng gần nhất (A)', 'Trailing 12 tháng gần nhất trước đó (A)']
sale = ti_le_tang_truong[0:4]
eps_p = ti_le_tang_truong[4:len(ti_le_tang_truong)]
list_nan = [0,0,0,0]
list_nan2 = [0,0]
sale = sale + list_nan
eps_p = list_nan + eps_p
tieu_chi = tieu_chi + tieu_chi
c_a = c[0:2] + a[0:2] + c[2:len(c)] + a[2:len(a)]

df_canslim = pd.DataFrame(tieu_chi, columns=['TieuChi'])
df_canslim['Sale'] = sale
df_canslim['EPS'] = eps_p
df_canslim['Diem'] = c_a

idf_canslim = df_canslim.interactive()

#point_slider = pn.widgets.FloatSlider(name='Tỉ lệ tăng trưởng', start=0, end=max(c_a), step=0.1, value=max(c_a))

yaxis_canslim = pn.widgets.RadioButtonGroup(
    name='Y axis',
    options=['Sale', 'EPS'],
    button_type='success'
)

canslim_pipeline = (
    idf_canslim[
        #(idf_canslim.Diem <= point_slider) &
        (idf_canslim.TieuChi.isin(tieu_chi))
    ]
    .groupby(['TieuChi', 'Diem'])[yaxis_canslim].mean()
    .to_frame()
    .reset_index()
    .sort_values(by='Diem')
    .reset_index(drop=True)
)

canslim_plot = canslim_pipeline.hvplot(x='Diem', by='TieuChi', y=yaxis_canslim, line_width=2, title="Canslim: "+str(round(sum(c_a),2))+"/100", stacked=True)
#canslim_pipeline_table = canslim_pipeline.pipe(pn.widgets.Tabulator, pagination='remote', page_size = 10, sizing_mode='stretch_width')

```

```

: df_canslim['Diem']

: 0    7.098497
  1    3.447551
  2    4.462469
  3    1.690610
  4    0.000000
  5    0.000000
  6    0.000000
  7    0.000000
Name: Diem, dtype: float64

```

```

10]: loai_tieu_chi = []
    for i in range(0,9):
        if(i>4):
            loai_tieu_chi.append('Tiêu chí EPS')
        elif(i==4):
            loai_tieu_chi.append('')
        else:
            loai_tieu_chi.append('Tiêu chí SALE')
    df_canslim_table = pd.DataFrame(loai_tieu_chi, columns=[''])
    df_canslim_table[''] = tieuchi[0:4]+''+tieuchi[4:len(tieuchi)]
    col_null = ''
    col_null_data = ['','','','','','','','']
    for i in range(0,8):
        df_canslim_table[col_null] = col_null_data
        col_null += ''
    df_canslim_table['Tỉ lệ tăng trưởng'] = ti_le_tang_truong[0:4]+''+ti_le_tang_truong[4:len(ti_le_tang_truong)]
    df_canslim_table['C'] = c[0:2] + ['','',''] + c[2:len(c)] + ['','','']
    df_canslim_table['A'] = a[0:2] + ['','',''] + a[2:len(a)]
    df_canslim_table['Tổng điểm'] = ['','','',round(sum(c_a),2),'','','']
    col_canslim = list(col_canslim)
    try:
        col_canslim.remove('Unnamed: 0')
    except:
        col_canslim = col_canslim

    df_canslim_table.iloc[0,5] = col_canslim[len(col_canslim)-5]+' '+str(sales[len(sales)-5])
    df_canslim_table.iloc[0,6] = col_canslim[len(col_canslim)-1]+' '+str(sales[len(sales)-1])
    df_canslim_table.iloc[1,5] = col_canslim[len(col_canslim)-6]+' '+str(sales[len(sales)-6])
    df_canslim_table.iloc[1,6] = col_canslim[len(col_canslim)-2]+' '+str(sales[len(sales)-2])

    df_canslim_table.iloc[5,5] = col_canslim[len(col_canslim)-5]+' '+str(eps[len(eps)-5])
    df_canslim_table.iloc[5,6] = col_canslim[len(col_canslim)-1]+' '+str(eps[len(eps)-1])
    df_canslim_table.iloc[6,5] = col_canslim[len(col_canslim)-6]+' '+str(eps[len(eps)-6])
    df_canslim_table.iloc[6,6] = col_canslim[len(col_canslim)-2]+' '+str(eps[len(eps)-2])
    for i in range(2,10):
        df_canslim_table.iloc[2,i] = col_canslim[i-1]+' '+str(sales[i-1])
        df_canslim_table.iloc[3,i] = col_canslim[i-2]+' '+str(sales[i-2])
        df_canslim_table.iloc[7,i] = col_canslim[i-1]+' '+str(eps[i-1])
        df_canslim_table.iloc[8,i] = col_canslim[i-2]+' '+str(eps[i-2])

    canslim_pipeline_table = df_canslim_table.pipe(pn.widgets.Tabulator, pagination='remote', page_size = 10, sizing_mode='stretch_width')

```

Hình 3.1.6: Trực quan hóa kết quả

3.2 Tính 4M

Bước 1: Import dữ liệu doanh nghiệp

```

df_CTKD = pd.read_csv("Data/"+business_code+"/CTKD_nam.csv", encoding='utf-16', sep='\t')
df_CDKT = pd.read_csv("Data/"+business_code+"/CDKT_nam.csv", encoding='utf-16', sep='\t')
df_CSTC = pd.read_csv("Data/"+business_code+"/CSTC_nam.csv", encoding='utf-16', sep='\t')

```

Hình 3.2.1: Import dữ liệu doanh nghiệp

Bước 2: Tiến hành tiền xử lý dữ liệu

2. Tiền xử lí dữ liệu

```
: df_CTKD.drop('Unnamed: 1', axis=1, inplace=True)
df_CTKD.drop('Unnamed: 2', axis=1, inplace=True)
df_CTKD.drop('Unnamed: 3', axis=1, inplace=True)

df_CDKT.drop('Unnamed: 1', axis=1, inplace=True)
df_CDKT.drop('Unnamed: 2', axis=1, inplace=True)
df_CDKT.drop('Unnamed: 3', axis=1, inplace=True)

df_CSTC.drop('Unnamed: 1', axis=1, inplace=True)
df_CSTC.drop('Unnamed: 2', axis=1, inplace=True)
df_CSTC.drop('Unnamed: 3', axis=1, inplace=True)

df_CTKD = df_CTKD.fillna(0)
df_CDKT = df_CDKT.fillna(0)
df_CSTC = df_CSTC.fillna(0)
```

Hình 3.2.2: Tiền xử lí dữ liệu

Bước 3: Khởi tạo các mảng dữ liệu và các hàm tính toán cần thiết

```
diem_tp = []

def chuyen_doi_float(type_data):
    type_data_dict = list(type_data)

    try:
        while(1):
            type_data_dict.remove('')
    except:
        type_data_dict = type_data_dict

    try:
        while(1):
            type_data_dict.remove('VND')
    except:
        type_data_dict = type_data_dict

    type_data = []

    for i in range(0, len(type_data_dict)):
        type_data.append(float(type_data_dict[i].replace(',', '')))
    return type_data
```

```

def calc(rate_something, type_data, tham_chieu, ty_trong):
    k = 1
    for i in range(0,3):
        rate_something.append(round(npf.rate(k,0,-(type_data[len(type_data) - k -2]),(type_data[len(type_data)-2]))*100,0))
        k+=2

    sum_tp = 0
    for i in range(0,len(rate_something)):
        if(rate_something[i] >= tham_chieu):
            sum_tp += 30
        else:
            if(i==1):
                sum_tp += (rate_something[i]/tham_chieu)*40
            else:
                sum_tp += (rate_something[i]/tham_chieu)*30
    #sum_tp = sum_tp*(ty_trong/100)
    diem_tp.append(sum_tp*(ty_trong/100))

def cac_rate(rate_something, type_data, tham_chieu, ty_trong):
    #f(type_data) != List):
    type_data = chuyen_doi_float(type_data)

    calc(rate_something, type_data, tham_chieu, ty_trong)
    return type_data

```

Hình 3.2.3: Khởi tạo mảng và hàm cần thiết

Bước 4: Tính toán các chỉ số

```

rate_sale = []
sale = df_CTKD.loc[6] #Doanh thu thuần hoạt động kinh doanh bảo hiểm
sale = list(sale)
sale.remove('Doanh thu thuần hoạt động kinh doanh bảo hiểm')
sale = cac_rate(rate_sale, sale, 20, 15)

rate_eps = []
eps = df_CSTC.loc[6] #Thu nhập trên mỗi cổ phần của 4 quý gần nhất (EPS)
eps = list(eps)
eps.remove('Thu nhập trên mỗi cổ phần của 4 quý gần nhất (EPS)')
eps = cac_rate(rate_eps, eps, 20, 20)

rate_bvps = []
bvps = df_CSTC.loc[7] #Giá trị sổ sách của cổ phiếu (BVPS)
bvps = list(bvps)
bvps.remove('Giá trị sổ sách của cổ phiếu (BVPS)')
bvps = cac_rate(rate_bvps, bvps, 15, 5)

rate_opc = []
opc = df_CTKD.loc[17] #Lưu chuyển tiền thuần từ hoạt động kinh doanh
opc = list(opc)
opc.remove('Lưu chuyển tiền thuần từ hoạt động kinh doanh')
opc_convert = []
for i in range(0, len(opc)):
    opc_convert.append(opc[i].replace('.', ''))
opc = cac_rate(rate_opc, opc_convert, 15, 15)

```

```

rate_effectiveness = []
tai_san = df_CDKT.loc[19] #Tổng cộng tài sản
tai_san = list(tai_san)
tai_san.remove('Tổng cộng tài sản')
tai_san = chuyen_doi_float(tai_san)
effectiveness = []
for i in range(0,len(tai_san)):
    effectiveness.append(sale[i]/tai_san[i])
calc(rate_effectiveness,effectiveness,10,5)

rate_efficiency = []
loi_nhuan = df_CTKD.loc[14] #Lợi nhuận sau thuế thu nhập doanh nghiệp
loi_nhuan = list(loih_nhuan)
loi_nhuan.remove('Lợi nhuận sau thuế thu nhập doanh nghiệp')
loi_nhuan = chuyen_doi_float(loih_nhuan)
efficiency = []
for i in range(0,len(loih_nhuan)):
    efficiency.append(loih_nhuan[i]/sale[i])
calc(rate_efficiency,efficiency,10,5)

rate_productivity = []
productivity = []
for i in range(0,len(loih_nhuan)):
    productivity.append(opc[i]/loih_nhuan[i])
calc(rate_productivity,productivity,10,5)

rate_roa = []
von = df_CDKT.loc[29] #Tổng cộng nguồn vốn
von = list(von)
von.remove('Tổng cộng nguồn vốn')
von = chuyen_doi_float(von)
roa = []
for i in range(0,len(loih_nhuan)):
    roa.append((loih_nhuan[i]/von[i])*100)
calc(rate_roa,roa,15,10)

rate_roe = []
von_chu_so_huu = df_CDKT.loc[24] #Vốn chủ sở hữu
von_chu_so_huu = list(von_chu_so_huu)
von_chu_so_huu.remove('Vốn chủ sở hữu')
von_chu_so_huu = chuyen_doi_float(von_chu_so_huu)
roe = []
for i in range(0,len(loih_nhuan)):
    roe.append((loih_nhuan[i]/von_chu_so_huu[i])*100)
calc(rate_roe,roe,20,5)

rate_roic = []
loi_nhuan_ctyme = df_CTKD.loc[15] #Lợi nhuận sau thuế của cổ đông Công ty mẹ
loi_nhuan_ctyme = list(loih_nhuan_ctyme)
loi_nhuan_ctyme.remove('Lợi nhuận sau thuế của cổ đông Công ty mẹ')
loi_nhuan_ctyme = chuyen_doi_float(loih_nhuan_ctyme)
roic = []
for i in range(0,len(loih_nhuan_ctyme)):
    roic.append((loih_nhuan_ctyme[i]/von[i])*100)
calc(rate_roic,roic,15,15)

```

Hình 3.2.4: Tính toán các chỉ số

Bước 5: Trực quan hóa kết quả

```
chi_so = ['Sales','EPS','BVPS','OPC LCDTKD','Effectiveness','Efficiency','Productivity','ROA','ROE','ROIC']
ty_trong_tp = [15,20,5,15,5,5,5,10,5,15]
df_4m = pd.DataFrame({
    'group': ['Tỷ trọng','Điểm thành phần'],
    'Sales Growth Rate': [ty_trong_tp[0], diem_tp[0]],
    'EPS Growth Rate': [ty_trong_tp[1], diem_tp[1]],
    'BVPS Growth Rates': [ty_trong_tp[2], diem_tp[2]],
    'Tăng trưởng OPC LCDTKD': [ty_trong_tp[3], diem_tp[3]],
    'Nợ dài hạn năm gần nhất': [10, 10],
    'Effectiveness': [ty_trong_tp[4], diem_tp[4]],
    'Efficiency': [ty_trong_tp[5], diem_tp[5]],
    'Productivity': [ty_trong_tp[6], diem_tp[6]],
    'ROA': [ty_trong_tp[7], diem_tp[7]],
    'ROE': [ty_trong_tp[8], diem_tp[8]],
    'ROIC': [ty_trong_tp[9], diem_tp[9]],
})
def visualization_4m():
    categories=list(df_4m)[1:]
    N = len(categories)

    angles = [n / float(N) * 2 * pi for n in range(N)]
    angles += angles[:1]

    ax = plt.subplot(111, polar=True)
    ax.set_theta_offset(pi / 2)
    ax.set_theta_direction(-1)

    plt.xticks(angles[:-1], categories)

    ax.set_rlabel_position(0)
    plt.yticks([-20,-15,-10,-5,0,5,10,15,20], ["-20","-15","-10","-5","0","5","10","15","20"], color="grey", size=7)
    plt.ylim(-20,20)
```

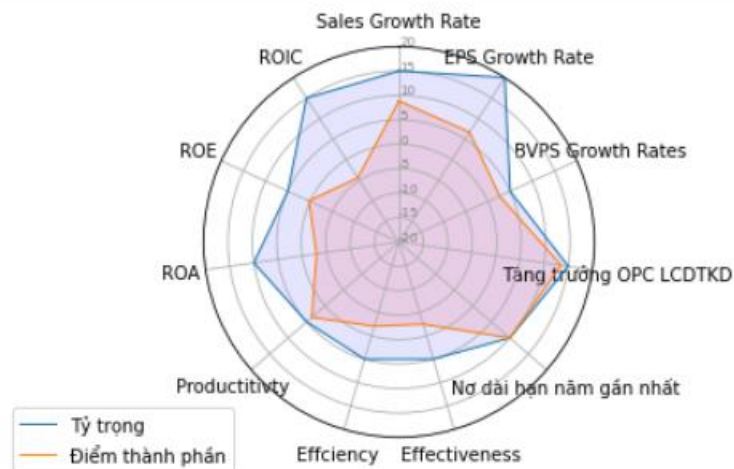
```
values=df_4m.loc[0].drop('group').values.flatten().tolist()
values += values[:1]
ax.plot(angles, values, linewidth=1, linestyle='solid', label="Tỷ trọng")
ax.fill(angles, values, 'b', alpha=0.1)

values=df_4m.loc[1].drop('group').values.flatten().tolist()
values += values[:1]
ax.plot(angles, values, linewidth=1, linestyle='solid', label="Điểm thành phần")
ax.fill(angles, values, 'r', alpha=0.1)

plt.legend(loc='upper right', bbox_to_anchor=(0.1, 0.1))
plt.savefig(business_code+'.png', bbox_inches='tight')
#return plt.show()
```



```
[16]: import os
if os.path.exists(business_code+'.png'):
    os.remove(business_code+'.png')
visualization_4m()
```



```
chiso=chi_so+chi_so+chi_so
df_4m2 = pd.DataFrame(chiso, columns=["ChiSo"])
null_list = [0,0,0,0,0,0,0,0,0,0,0,0]
nam_list1 = [1,1,1,1,1,1,1,1,1,1,1,1]
nam_list3 = [3,3,3,3,3,3,3,3,3,3,3,3]
nam_list5 = [5,5,5,5,5,5,5,5,5,5,5,5]
nam_1 = [rate_sale[0],rate_eps[0],rate_bvps[0],rate_opc[0],rate_effectiveness[0],rate_efficiency[0],rate_productivity[0],rate_roa[0],rate_roe[0],rate_roic[0]]
nam_3 = [rate_sale[1],rate_eps[1],rate_bvps[1],rate_opc[1],rate_effectiveness[1],rate_efficiency[1],rate_productivity[1],rate_roa[1],rate_roe[1],rate_roic[1]]
nam_5 = [rate_sale[2],rate_eps[2],rate_bvps[2],rate_opc[2],rate_effectiveness[2],rate_efficiency[2],rate_productivity[2],rate_roa[2],rate_roe[2],rate_roic[2]]
df_4m2['Nam'] = nam_list1 + nam_list3 + nam_list5
df_4m2['TrongSo'] = nam_1 + nam_3 + nam_5

idf_4m = df_4m2.interactive()

year_slider = pn.widgets.IntSlider(name='Year slider', start=1, end=5, step=2, value=1)

vis4m_pipeline = (
    idf_4m[
        (idf_4m.Nam <= year_slider) &
        (idf_4m.ChiSo.isin(chi_so))
    ]
    .groupby(['ChiSo', 'Nam'])['TrongSo'].mean()
    .to_frame()
    .reset_index()
    .sort_values(by='Nam')
    .reset_index(drop=True)
)

vis4m_plot = vis4m_pipeline.hvplot(x='Nam', by='ChiSo', line_width=2, title="4M: "+str(sum(diem_tp)+10)+'/100', rot=90)
```

Hình 3.2.5: Trực quan hóa kết quả

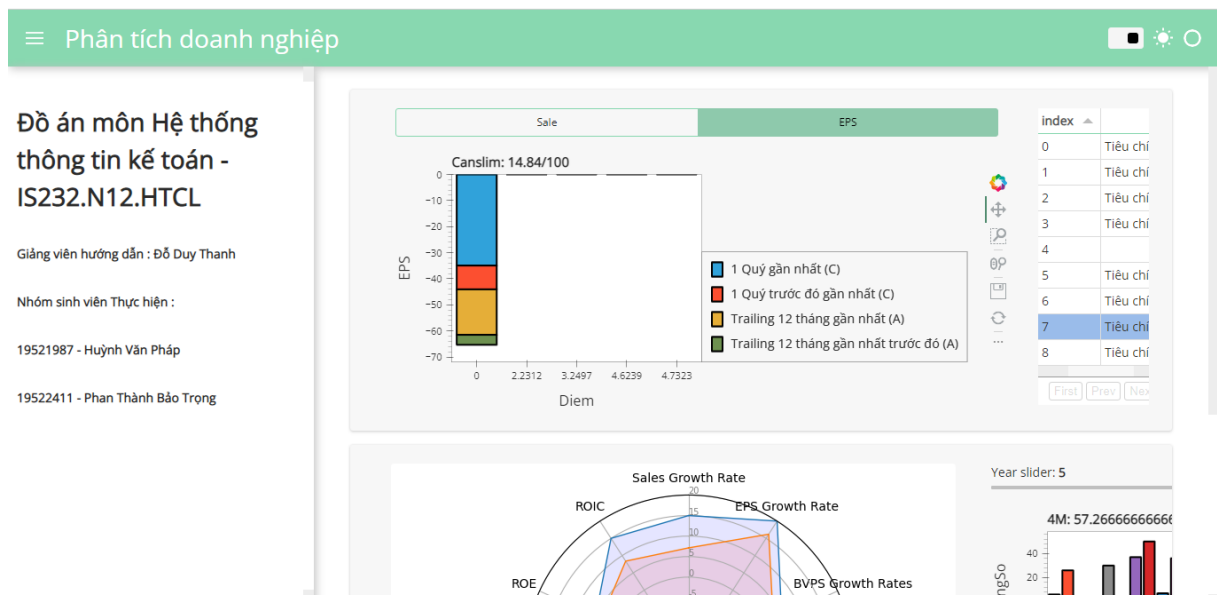
3.3 Khởi tạo server local hiển thị Dashboard

```

template = pn.template.FastListTemplate(
    title='Phân tích doanh nghiệp',
    sidebar=[pn.pane.Markdown("# Đồ án môn Hệ thống thông tin kế toán - IS232.N12.HTCL"),
        pn.pane.Markdown("#### Giảng viên hướng dẫn : Đỗ Duy Thanh"),
        pn.pane.Markdown("#### Nhóm sinh viên Thực hiện :"),
        pn.pane.Markdown("#### 19521987 - Huỳnh Văn Pháp"),
        pn.pane.Markdown("#### 19522411 - Phan Thành Bảo Trọng")
    ],
    main=[pn.Row(pn.Column(yaxis_canslim,
        canslim_plot.panel(width=700), margin=(0,25)),
        canslim_pipeline_table),
        pn.Row(pn.Column(pn.pane.PNG(business_code+'.png', sizing_mode='scale_both'), margin=(0,25)),
        pn.Column(year_slider, vis4m_plot.panel(width=600)))],
    accent_base_color="#88d8b0",
    header_background="#88d8b0",
)
template.servable();

```

Hình 3.3.1: Khởi tạo server local



Hình 3.3.2: Kết quả sau khi chạy chương trình

Chương 4 KẾT LUẬN

4.1 Ưu điểm

- Học được nhiều kiến thức mới Canslim và 4M
- Đọc được nhiều bài viết hay về Canslim và 4M
- Trau dồi được thêm nhiều kiến thức về ngôn ngữ lập trình python
- Tăng khả năng làm việc nhóm

4.2 Nhược điểm

- Tốn nhiều thời gian để nghiên cứu và tìm hiểu
- Khó khăn trong quá trình nghiên cứu và học hỏi kiến thức mới
- Khó khăn trong việc sắp xếp thời gian dự án giữa các thành viên

4.3 Hướng phát triển

- Mở rộng dữ liệu doanh nghiệp
- Phát triển thêm nhiều tính năng đánh giá khác
- Cải thiện giao diện và thuật toán, giúp cho ứng dụng chạy nhanh hơn và hiệu quả hơn
- Cải thiện ứng dụng, tăng độ chính xác kết quả đầu ra

TÀI LIỆU THAM KHẢO

- [1] <https://iife.edu.vn/tim-hieu-nguyen-tac-dau-tu-canslim/>
- [2] <https://happy.live/khi-4m-ket-hop-cung-canslim/>
- [3] <https://vi.wikipedia.org/wiki/CANSLIM>
- [4] <https://finance.vietstock.vn/>
- [5] https://www.youtube.com/watch?v=uhxiXOTKzfs&ab_channel=ThuVu%20dataanalytics
- [6] <https://www.youtube.com/watch?v=hSPmj7mK6ng>