

TEAM
MHT

CODING ARGUMENTATION

SC HACKFEST BY SCH

OUR TEAM MEMBER



HOÀNG THÁI MINH

Sinh viên K60
Logistics và quản lý chuỗi cung ứng
Đại học Ngoại thương



NGUYỄN THỊ KHÁNH HUYỀN

Sinh viên K60
Logistics và quản lý chuỗi cung ứng
Đại học Ngoại thương



TRỊNH PHÚ THĂNG

Sinh viên K60
Kinh doanh quốc tế theo mô
hình Nhật Bản
Đại học Ngoại thương

CONTENT LIST

01

BUSINESS UNDERSTANDING

Xác định những yếu tố quan trọng và phương thức thành công của dự án

02

ANALYTIC APPROACH

Hướng xử lý dữ liệu để giải quyết

03

DATA ACQUISITION AND UNDERSTANDING

Chuẩn bị cơ sở dữ liệu phù hợp, hiệu quả. Phát triển phương pháp làm mới và mở rộng dữ liệu

04

MODELING

Tạo ra sự liên kết giữa dữ liệu, tạo ra model để giải quyết vấn đề

BUSINESS UNDERSTANDING

FESTORY

Festory là nhà máy chuyên sản xuất thức ăn chăn nuôi có trụ sở chính tại Bình Dương, trong đó thức ăn chăn nuôi cho heo và cho cá là 2 sản phẩm chính đem lại nguồn doanh thu lớn cho phía nhà máy.

QUY ĐỊNH

Festory hiện chỉ vận hành 1 kho duy nhất đặt tại nhà máy để lưu trữ cả nguyên vật liệu (raw material) và sản phẩm (finished goods). kho của Festory hiện đang được quản lý và vận hành theo nguyên tắc FEFO (First Expired - First Out).



DỰ ĐOÁN NHU CẦU

Festory hiện đang duy trì hoạt động từ Thứ 2 đến Chủ nhật và mỗi ngày đều có 2 ca làm. Nhu cầu cho 2 sản phẩm trên là tương đối ổn định và có thể dự đoán được, cụ thể, dự đoán nhu cầu trong 6 tháng đầu năm của thức ăn cho heo và thức ăn cho cá được cung cấp như sau:

ITEM	JAN_2023	FEB_2023	MAR_2023	APR_2023	MAY_2023	JUN_2023
THỨC_ĂN_CHO_HEO	18	22	30	25	24	27
THỨC_ĂN_CHO_CÁ	16	21	22	23	23	26

CHI TIẾT NGUYÊN LIỆU

Festory tập trung sản xuất 2 dòng sản phẩm chính là Thức ăn cho heo và thức ăn cho cá (hao hụt trong sản xuất không đáng kể). Chi tiết nguyên liệu ở bảng sau:

THÀNH PHẨM	NGUYÊN LIỆU	SỐ LƯỢNG NGUYÊN LIỆU (TRÊN 1 TẤN THÀNH PHẨM)	ĐƠN VỊ CỦA NGUYÊN LIỆU
THỨC_ĂN_CHO_HEO	NGÔ	400	kg
THỨC_ĂN_CHO_HEO	KHOAI	200	kg
THỨC_ĂN_CHO_HEO	SẮN	200	kg
THỨC_ĂN_CHO_HEO	XƯƠNG_ĐỘNG_VẬT	200	kg
THỨC_ĂN_CHO_HEO	MEN_VI_SINH	10	lít
THỨC_ĂN_CHO_HEO	BAO_BÌ_ĐÓNG_GÓI	40	cái
THỨC_ĂN_CHO_CÁ	NGÔ	500	kg
THỨC_ĂN_CHO_CÁ	BỘT_ĐỖ_TƯƠNG	300	kg
THỨC_ĂN_CHO_CÁ	THÓC	100	kg
THỨC_ĂN_CHO_CÁ	XƯƠNG_ĐỘNG_VẬT	100	kg
THỨC_ĂN_CHO_CÁ	MEN_VI_SINH	10	lít
THỨC_ĂN_CHO_CÁ	BAO_BÌ_ĐÓNG_GÓI	40	cái



XÁC ĐỊNH MỤC TIÊU

Bài toán đặt ra mục tiêu để giải quyết là: xác định số lượng nguyên liệu cần đặt hàng và các thời điểm đặt hàng để đảm bảo cung ứng đủ cho khách hàng theo dự báo và đáp ứng các yêu cầu đưa ra từ Ban Giám đốc, bao gồm: kho được vận hành theo nguyên tắc:

- FEFO (First Expire, First Out)
- Số ngày tồn kho trung bình của các mặt hàng tối đa là 30 ngày
- Đặt nguyên liệu theo quy tắc MOQ (Minimum order quantity)
- Nguyên liệu được coi là hết hạn (không sử dụng được nữa) khi ngày hiện tại trùng hoặc lớn hơn hạn sử dụng của nguyên liệu

DATA SOURCES



Data sẽ được lấy trực tiếp từ bộ Data đội thi được cung cấp trong file [DATA ROUND2 SCHACKFEST2022.xlsx](#), bao gồm các mục sau:

BẢNG 1: Dự đoán nhu cầu cho thức ăn cho heo và thức ăn cho cá 6 tháng đầu năm 2023
(đơn vị: tấn)

BẢNG 2: Thông tin nguyên liệu (BOM) của thức ăn cho heo và thức ăn cho cá

BẢNG 3: Thông tin nguyên liệu trong kho (Stock on hand) vào cuối tháng 12

BẢNG 4: Thông tin nguyên liệu đang trên đường về
(Ongoing supply)

BẢNG 5: Thông tin bổ sung về leadtime và shelf life của các nguyên liệu

BẢNG 6: Thông tin bổ sung về leadtime và shelf life của thành phẩm

ANALYTIC APPROACH



Đội thi tiếp cận bài toán theo cách phân tích đề xuất (Prescriptive approach), tức là sử dụng nguồn data đã có về nhu cầu, nguồn cung sẵn có (stock on hand), hạn sử dụng của các loại mặt hàng để tìm ra số lượng nguyên liệu cần mua cũng như thời điểm đặt hàng để đảm bảo nguồn nguyên liệu đủ và sản xuất đủ thành phẩm cho nhu cầu của khách hàng.

Supply Chain Analytics Capabilities For Specialty Retailers

1	Descriptive	>	How much did I sell where? How much stock am I invested in right now? Where did I stock out? Where are my holes in the store assortment? How much did the competition sell a similar item for? Where and how far am I behind in my sales targets? How many online clicks did I get and how many did I convert? What was my average price or GMROI per sub-category and market?
2	Diagnostic	>	Why did my sales decrease in a certain sub-segment? What are the possible reasons for stock out in a given region? How did I end up with stock excess at the end of the season? Why are my GMROI numbers different for similar product clusters?
3	Predictive	>	What is my life-cycle forecast for my summer portfolio of products? What is the likely lift in my price promotion in this sub-category? What are the likely trends in competitors' pricing? Which products are likely to stock out in the next two weeks? For which SKUs am I likely to end up with end-of-season excess? What might happen to my stock position if I reduce my price by 20%?
4	Prescriptive	>	How should I segment my stores? What should my assortment? How many should I buy/procure in pre-season? At what price? What percentage of my initial buy should I send to stores directly? How much stock should I allocate for on-line vs off-line sales? How much safety stock should I keep in each SKU at each store? How frequently should I replenish a store? When and how much should I replenish/procure for in-season planning?



DATA ACQUISITION AND UNDERSTANDING

Data liên quan tới bên cung cấp nguyên liệu cũng như dự báo về nhu cầu hàng hóa được cung cấp sẵn. Một sheet excel bao gồm: DEMAND, BOM (Bill of Material), SUPPLY-SOH (Stock on hand), SUPPLY-ONGOING (Ongoing Delivery), ITEM MASTER - RAW, ITEM MASTER - FG. Từ đó, team đã có đủ lượng dữ liệu cần thiêt và phù hợp để giải quyết vấn đề.

MODELING

Bài làm của MHT chia làm 5 bước chính:



BƯỚC 1:

Đọc dữ liệu

BƯỚC 3:

Tạo df_supply sử dụng trong
kỳ đầu tiên từ
df_SUPPLY_SOH và
df_SUPPLY_ONGOING

BƯỚC 5:

Xuất ra file kết quả

BƯỚC 2:

Tìm số lượng nguyên liệu dự
đoán theo nhu cầu dự đoán

BƯỚC 4:

Dùng vòng lặp để duyệt qua từng
kỳ để tìm số lượng nguyên liệu
đặt hàng theo MOQ

BƯỚC 1:

Đọc dữ liệu đầu vào cho các dataframe tương ứng

```
In [2]: 1 sheet_id = '1FLY9x5xmQASC4b9DQeMaEH_21PNK-hw7'
2 sheet_name = '1_DEMAND'
3 url = f"https://docs.google.com/spreadsheets/d/{sheet_id}/gviz/tq?tqx=out:csv&sheet={sheet_name}"
4 df_demand = pd.read_csv(url)
5 # đơn vị thành phẩm: tấn
```

```
In [3]: 1 sheet_name = '2_BOM'
2 url = f"https://docs.google.com/spreadsheets/d/{sheet_id}/gviz/tq?tqx=out:csv&sheet={sheet_name}"
3 df_BOM = pd.read_csv(url, usecols=[0,1,2])
4 # đơn vị là kg (ngô, khoai, sắn, xương động vật, Bột đồ tương, thóc), cái (bao bì), lít (men vi sinh)
```

```
In [4]: 1 sheet_name = '3_SUPPLY-SOH'
2 url = f"https://docs.google.com/spreadsheets/d/{sheet_id}/gviz/tq?tqx=out:csv&sheet={sheet_name}"
3 df_SUPPLY_SOH = pd.read_csv(url, usecols=[0,1,3])
4 # đơn vị là kg (ngô, khoai, sắn, xương động vật, Bột đồ tương, thóc), cái (bao bì), lít (men vi sinh)
5
6 # converting datetime
7 df_SUPPLY_SOH['HẠN SỬ DỤNG'] = pd.to_datetime(df_SUPPLY_SOH['HẠN SỬ DỤNG'], format='%d-%b-%Y')
```

```
In [5]: 1 sheet_name = '4_SUPPLY-ONGOING'
2 url = f"https://docs.google.com/spreadsheets/d/{sheet_id}/gviz/tq?tqx=out:csv&sheet={sheet_name}"
3 df_SUPPLY_ONGOING = pd.read_csv(url, usecols=[0,2,3,5])
4 # đơn vị là kg (ngô, khoai, sắn, xương động vật, Bột đồ tương, thóc), cái (bao bì), lít (men vi sinh)
5
6 # converting datetime
7 df_SUPPLY_ONGOING['NGÀY VỀ'] = pd.to_datetime(df_SUPPLY_ONGOING['NGÀY VỀ'], format='%d-%b-%Y')
8 df_SUPPLY_ONGOING['HẠN SỬ DỤNG'] = pd.to_datetime(df_SUPPLY_ONGOING['HẠN SỬ DỤNG'], format='%d-%b-%Y')
```

```
In [6]: 1 sheet_name = '5_ITEM MASTER - RAW'
2 url = f"https://docs.google.com/spreadsheets/d/{sheet_id}/gviz/tq?tqx=out:csv&sheet={sheet_name}"
3 url = url.replace(" ", "%20")
4 df_ITEM_MASTER_RAW = pd.read_csv(url, usecols=[0,2,3,4,6])
5 df_ITEM_MASTER_RAW.rename(columns={'ITEM': 'NGUYÊN LIỆU'}, inplace=True)
6 df_ITEM_MASTER_RAW.fillna(365, inplace=True)
```

```
In [7]: 1 sheet_name = '6_ITEM MASTER - FG'
2 url = f"https://docs.google.com/spreadsheets/d/{sheet_id}/gviz/tq?tqx=out:csv&sheet={sheet_name}"
3 url = url.replace(" ", "%20")
4 df_ITEM_MASTER_FG = pd.read_csv(url, usecols=[0,2,3])
```

Bước 2: Tìm số lượng nguyên liệu dự đoán theo nhu cầu dự đoán

2.1. Đẩy lượng hàng còn tồn trong kho từ cuối tháng 12 vào demand của tháng 1

```
[ ] # Lượng hàng thành phẩm còn tồn trong kho từ cuối tháng 12 sẽ được dùng để cung ứng cho tháng đầu tiên theo quy định số ngày tồn kho trung bình tối đa  
# merge df_demand với df_SUPPLY_SOH để tìm số lượng hàng thành phẩm đang có trong kho  
df_demand = df_demand.merge(df_SUPPLY_SOH[['ITEM', 'TỒN KHO CUỐI NGÀY 31/12/2022']], on='ITEM', how='left')  
  
# số lượng hàng thành phẩm còn tồn trong kho sẽ được trừ trực tiếp vào demand của tháng 1  
df_demand['JAN_2023'] = df_demand['JAN_2023'] - df_demand['TỒN KHO CUỐI NGÀY 31/12/2022']/1000  
  
# xóa cột 'TỒN KHO CUỐI NGÀY 31/12/2022'  
df_demand.drop(columns=['TỒN KHO CUỐI NGÀY 31/12/2022'], inplace=True)  
df_demand
```

	ITEM	JAN_2023	FEB_2023	MAR_2023	APR_2023	MAY_2023	JUN_2023
0	THỨC ĂN CHO HEO	11.0	22	30	25	24	27
1	THỨC ĂN CHO CÁ	8.0	21	22	23	23	26

Bước 2: Tìm số lượng nguyên liệu dự đoán theo nhu cầu dự đoán

2.2. Unpivot df_demand để tính toán

```
# unpivot df_demand
df_demand_melt = df_demand.melt('ITEM')
df_demand_melt = df_demand_melt.rename(columns = {'ITEM':'THÀNH PHẨM',
                                                 'variable':'forecast_time',
                                                 'value':'monthly_demand_qty'})

# converting datetime
df_demand_melt['forecast_time'] = pd.to_datetime(df_demand_melt['forecast_time'], format='%b_%Y')

# sắp xếp lại theo 'THÀNH PHẨM'
df_demand_melt = df_demand_melt.sort_values('forecast_time', ignore_index=True)
df_demand_melt
```

Bảng dự đoán nhu cầu hàng tháng:

	THÀNH PHẨM	forecast_time	monthly_demand_qty
0	THỨC ĂN CHO HEO	2023-01-01	11.0
1	THỨC ĂN CHO CÁ	2023-01-01	8.0
2	THỨC ĂN CHO HEO	2023-02-01	22.0
3	THỨC ĂN CHO CÁ	2023-02-01	21.0
4	THỨC ĂN CHO HEO	2023-03-01	30.0
5	THỨC ĂN CHO CÁ	2023-03-01	22.0
6	THỨC ĂN CHO HEO	2023-04-01	25.0
7	THỨC ĂN CHO CÁ	2023-04-01	23.0
8	THỨC ĂN CHO HEO	2023-05-01	24.0
9	THỨC ĂN CHO CÁ	2023-05-01	23.0
10	THỨC ĂN CHO HEO	2023-06-01	27.0
11	THỨC ĂN CHO CÁ	2023-06-01	26.0

Bước 2: Tìm số lượng nguyên liệu dự đoán theo nhu cầu dự đoán

2.3 Tìm sản lượng sản xuất theo từng ngày dựa theo df_demand (giả thiết: để đáp ứng nhu cầu của tháng đó thì mỗi ngày công ty sẽ sản xuất một lượng bằng nhu cầu/tổng số ngày trong tháng)

- Nhóm chọn cách tiếp cận xác định số lượng hàng thành phẩm sản xuất mỗi ngày bằng tổng nhu cầu của 1 tháng chia cho số ngày trong tháng. Hướng làm này giúp xác định chính xác thời gian các nguyên liệu bị thiếu để chọn thời gian đặt hàng tối ưu nhất.

```
# tạo df_daily_production dựa theo df_demand_melt
df_daily_production = df_demand_melt
df_daily_production = df_daily_production.rename(columns={'monthly_demand_qty':'daily_production_qty'})

# tạo ra các ngày tương ứng trong từng tháng + thêm vào df_daily_production; tính toán daily_production_qty tương ứng
for i, data in df_demand_melt.iterrows():
    x = df_demand_melt.loc[i, 'forecast_time']
    df_daily_production.loc[i, 'daily_production_qty'] = df_demand_melt.loc[i, 'monthly_demand_qty'] / pd.Period(str(x)).days_in_month

    for j in range(pd.Period(str(x)).days_in_month - 1):
        df_daily_production.loc[len(df_daily_production)] = [df_demand_melt.loc[i,'THÀNH PHẨM'],
                                                x + datetime.timedelta(days = j + 1),
                                                df_demand_melt.loc[i, 'monthly_demand_qty'] / pd.Period(str(x)).days_in_month]

df_daily_production = df_daily_production.sort_values(by=[ 'THÀNH PHẨM', 'forecast_time'], ignore_index=True)
df_daily_production
```

	THÀNH PHẨM	forecast_time	daily_production_qty
0	THỨC ĂN CHO CÁ	2023-01-01	0.258065
1	THỨC ĂN CHO CÁ	2023-01-02	0.258065
2	THỨC ĂN CHO CÁ	2023-01-03	0.258065
3	THỨC ĂN CHO CÁ	2023-01-04	0.258065
4	THỨC ĂN CHO CÁ	2023-01-05	0.258065
...
357	THỨC ĂN CHO HEO	2023-06-26	0.900000
358	THỨC ĂN CHO HEO	2023-06-27	0.900000
359	THỨC ĂN CHO HEO	2023-06-28	0.900000
360	THỨC ĂN CHO HEO	2023-06-29	0.900000
361	THỨC ĂN CHO HEO	2023-06-30	0.900000

362 rows × 3 columns

Bước 2: Tìm số lượng nguyên liệu dự đoán theo nhu cầu dự đoán

2.4. Dựa vào leadtime sản xuất để xác định ngày bắt đầu sản xuất

- Trong khuôn khổ đề bài, leadtime sản xuất của 2 thành phẩm đều bằng 0 nhưng với tiêu chí áp dụng với các bộ data khác, leadtime sản xuất có thể là 1 ngày, 2 ngày, ... do đó bước này giúp xác định thời gian bắt đầu sản xuất hàng để đáp ứng đủ nhu cầu thành phẩm của từng tháng.

```
# merge df_daily_production với dataframe chứa leadtime ngày sản xuất
df_daily_production = df_daily_production.merge(df_ITEM_MASTER_FG[['SKU', 'LEADTIME SẢN XUẤT \n(NGÀY)']],
                                                left_on='THÀNH PHẨM',
                                                right_on='SKU',
                                                how='left')

# dùng vòng lặp để xác định số ngày bắt đầu sản xuất - job_start_date
for i, data in df_daily_production.iterrows():
    df_daily_production.loc[i, 'job_start_date'] = df_daily_production.loc[i, 'forecast_time'] - datetime.timedelta(days=int(df_daily_production.loc[i, 'LEADTIME SẢN XUẤT \n(NGÀY)']))

# Bỏ 3 cột 'forecast_time', 'SKU' và 'LEADTIME SẢN XUẤT \n(NGÀY)' khỏi dataframe
df_daily_production.drop(columns=['forecast_time', 'SKU', 'LEADTIME SẢN XUẤT \n(NGÀY)'], inplace=True)
df_daily_production
```

	THÀNH PHẨM	daily_production_qty	job_start_date
0	THỨC ĂN CHO CÁ	0.258065	2023-01-01
1	THỨC ĂN CHO CÁ	0.258065	2023-01-02
2	THỨC ĂN CHO CÁ	0.258065	2023-01-03
3	THỨC ĂN CHO CÁ	0.258065	2023-01-04
4	THỨC ĂN CHO CÁ	0.258065	2023-01-05
...
357	THỨC ĂN CHO HEO	0.900000	2023-06-26
358	THỨC ĂN CHO HEO	0.900000	2023-06-27
359	THỨC ĂN CHO HEO	0.900000	2023-06-28
360	THỨC ĂN CHO HEO	0.900000	2023-06-29
361	THỨC ĂN CHO HEO	0.900000	2023-06-30

Bước 2: Tìm số lượng nguyên liệu dự đoán theo nhu cầu dự đoán

2.5. Tìm số lượng nguyên liệu cần để sản xuất theo từng ngày theo từng thành phẩm

- Trong khuôn khổ đề bài, leadtime sản xuất của 2 thành phẩm đều bằng 0 nhưng với tiêu chí áp dụng với các bộ data khác, leadtime sản xuất có thể là 1 ngày, 2 ngày, ... do đó bước này giúp xác định thời gian bắt đầu sản xuất hàng để đáp ứng đủ nhu cầu thành phẩm của từng tháng.

```
# merge df_daily_production và df_BOM
df_daily_production_merged = df_daily_production.merge(df_BOM, how='right', on='THÀNH PHẨM', sort=True)

# Xác định số lượng nguyên liệu cần để sản xuất theo từng ngày
df_daily_production_merged['daily_demand_raw_qty'] = df_daily_production_merged['daily_production_qty'] * df_daily_production_merged['SỐ LƯỢNG NGUYÊN LIỆU \n(TRÊN 1 TẤN THÀNH PHẨM)']
df_daily_production_merged
```

	THÀNH PHẨM	daily_production_qty	job_start_date	NGUYÊN LIỆU	SỐ LƯỢNG NGUYÊN LIỆU \n(TRÊN 1 TẤN THÀNH PHẨM)	daily_demand_raw_qty
0	THỨC ĂN CHO CÁ	0.258065	2023-01-01	NGÔ	500	129.032258
1	THỨC ĂN CHO CÁ	0.258065	2023-01-02	NGÔ	500	129.032258
2	THỨC ĂN CHO CÁ	0.258065	2023-01-03	NGÔ	500	129.032258
3	THỨC ĂN CHO CÁ	0.258065	2023-01-04	NGÔ	500	129.032258
4	THỨC ĂN CHO CÁ	0.258065	2023-01-05	NGÔ	500	129.032258
...
2167	THỨC ĂN CHO HEO	0.900000	2023-06-26	BAO BÌ ĐÓNG GÓI	40	36.000000
2168	THỨC ĂN CHO HEO	0.900000	2023-06-27	BAO BÌ ĐÓNG GÓI	40	36.000000
2169	THỨC ĂN CHO HEO	0.900000	2023-06-28	BAO BÌ ĐÓNG GÓI	40	36.000000
2170	THỨC ĂN CHO HEO	0.900000	2023-06-29	BAO BÌ ĐÓNG GÓI	40	36.000000
2171	THỨC ĂN CHO HEO	0.900000	2023-06-30	BAO BÌ ĐÓNG GÓI	40	36.000000

2172 rows × 6 columns

Bước 2: Tìm số lượng nguyên liệu dự đoán theo nhu cầu dự đoán

2.6. Tìm tổng số lượng nguyên liệu cần để sản xuất theo từng ngày

- Vì thành phẩm được tạo thành từ một vài nguyên liệu giống nhau nên bước này dùng để tính tổng lượng nguyên liệu cần để sản xuất thành phẩm trong 1 ngày.

```
# cumsum 'daily_demand_raw_qty' theo từng ngày và theo 'NGUYÊN LIỆU'  
df_daily_production_merged['cumsum_daily_demand_raw_qty'] = df_daily_production_merged.groupby(by=[df_daily_production_merged['job_start_date'],  
                                              df_daily_production_merged['NGUYÊN LIỆU']])['daily_demand_raw_qty'].cumsum()  
  
# lấy giá trị cuối cùng khi groupby theo từng ngày và từng nguyên liệu để biết được tổng số lượng cần của 1 nguyên liệu trong 1 ngày  
df_daily_production_merged = df_daily_production_merged.groupby(by=[df_daily_production_merged['job_start_date'], df_daily_production_merged['NGUYÊN LIỆU']],  
                  group_keys=False,  
                  sort=False).last()  
  
# reset index của df_daily_production_merged  
df_daily_production_merged.reset_index(inplace=True)  
  
# loại bỏ những cột không còn sử dụng nữa trong df_daily_production_merged  
df_daily_production_merged.drop(columns=['THÀNH PHẨM',  
                                         'daily_production_qty',  
                                         'daily_demand_qty',  
                                         'SỐ LƯỢNG NGUYÊN LIỆU \n(TRÊN 1 TẤN THÀNH PHẨM)', inplace=True)  
  
df_daily_production_merged
```

	job_start_date	NGUYÊN LIỆU	cumsum_daily_demand_raw_qty
0	2023-01-01	NGÔ	270.967742
1	2023-01-02	NGÔ	270.967742
2	2023-01-03	NGÔ	270.967742
3	2023-01-04	NGÔ	270.967742
4	2023-01-05	NGÔ	270.967742
...
1443	2023-06-26	SÂN	180.000000
1444	2023-06-27	SÂN	180.000000
1445	2023-06-28	SÂN	180.000000
1446	2023-06-29	SÂN	180.000000
1447	2023-06-30	SÂN	180.000000

1448 rows × 3 columns

Bước 3: Tạo df_supply dùng cho kỳ đầu tiên từ df_SUPPLY_SOH và df_SUPPLY_ONGOING

3.1. Format df_SUPPLY_SOH

- Đặt 'NGÀY VỀ' của SOH là '2000-1-1' để đảm bảo logic theo FEFO: hạn sử dụng của các nguyên liệu trong kho chắc chắn sẽ nhỏ hơn hạn sử dụng của nguyên liệu đang trên đường về hoặc nguyên liệu order sau này.

```
[ ] # Đổi tên cột
df_SUPPLY_SOH.rename(columns={'TỒN KHO CUỐI NGÀY 31/12/2022':'SỐ LƯỢNG',
                             'ITEM':'NGUYÊN LIỆU'}, inplace=True)
# Thêm 1 cột là 'NGÀY VỀ' và giá trị là '2000-1-1' để đáp ứng mục đích sort_values
df_SUPPLY_SOH.insert(1, column='NGÀY VỀ', value=pd.to_datetime('2000-1-1'))
df_SUPPLY_SOH
```

	NGUYÊN LIỆU	NGÀY VỀ	SỐ LƯỢNG	HẠN SỬ DỤNG
0	KHOAI	2000-01-01	10000	2023-02-12
1	SÁN	2000-01-01	3000	2023-02-27
2	XƯƠNG_ĐỘNG_VẬT	2000-01-01	2000	2023-01-26
3	BỘT_ĐỒ_TƯƠNG	2000-01-01	9000	2023-02-20
4	MEN_VI_SINH	2000-01-01	200	2023-01-25
5	NGÔ	2000-01-01	12000	2023-02-02
6	THÓC	2000-01-01	13000	2023-03-03
7	BAO_BÌ_ĐÓNG_GÓI	2000-01-01	1220	NaT
8	THÚC_ĂN_CHO_HEO	2000-01-01	7000	2023-01-27
9	THÚC_ĂN_CHO_CÁ	2000-01-01	8000	2023-01-30

Bước 3: Tạo df_supply dùng cho kỳ đầu tiên từ df_SUPPLY_SOH và df_SUPPLY_ONGOING

3.2. Concat df_SUPPLY_SOH và df_SUPPLY_ONGOING

- Ghép 2 dataframe: df_SUPPLY và df_SUPPLY_ONGOING và sort_values

```
df_supply = pd.concat([df_SUPPLY_SOH, df_SUPPLY_ONGOING], axis=0)

# Sắp xếp dữ liệu theo 'ITEM' và theo 'NGÀY VỀ', không cần sắp xếp theo 'HẠN SỬ DỤNG' vì chắc chắn HSD của SOH luôn nhỏ hơn HSD của ONGOING SUPPLY
df_supply = df_supply.sort_values(by=[ 'NGUYÊN LIỆU', 'NGÀY VỀ'])

# Đặt HSD của BAO_BÌ_ĐÓNG_GÓI là '2100-1-1' vì BAO_BÌ_ĐÓNG_GÓI không có HSD
df_supply.fillna(pd.to_datetime('2100-1-1'), inplace=True)
df_supply
```

	NGUYÊN LIỆU	NGÀY VỀ	SỐ LƯỢNG	HẠN SỬ DỤNG
7	BAO_BÌ_ĐÓNG_GÓI	2000-01-01	1220	2100-01-01
3	BỘT_ĐỒ_TƯƠNG	2000-01-01	9000	2023-02-20
0	KHOAI	2000-01-01	10000	2023-02-12
0	KHOAI	2023-01-03	1800	2023-04-13
4	MEN_VI_SINH	2000-01-01	200	2023-01-25
5	NGÔ	2000-01-01	12000	2023-02-02
1	NGÔ	2023-01-05	1000	2023-04-15
1	SẮN	2000-01-01	3000	2023-02-27
2	SẮN	2023-01-12	1000	2023-04-22
6	THÓC	2000-01-01	13000	2023-03-03
9	THÚC_ĂN_CHO_CÁ	2000-01-01	8000	2023-01-30
8	THÚC_ĂN_CHO_HEO	2000-01-01	7000	2023-01-27
2	XƯƠNG_ĐỘNG_VẬT	2000-01-01	2000	2023-01-26

Bước 4: Duyệt qua từng kỳ để tìm số lượng nguyên liệu cần đặt theo MOQ

4.1. Tìm ra số kỳ đặt hàng

- Xác định số kỳ đặt hàng dựa theo số ngày tồn kho trung bình

```
# Tìm ra tổng số ngày sản xuất
total_days = df_daily_production_merged['job_start_date'].max() - df_daily_production_merged['job_start_date'].min()
total_days = total_days / datetime.timedelta(days=1)

# age_of_inventory = Số ngày tồn kho trung bình tối đa của các mặt hàng
# (Biến này có thể thay đổi tùy theo quy định của Ban giám đốc)
age_of_inventory = 30

# Tìm ra số kỳ đặt hàng: nper
nper = math.ceil(total_days/age_of_inventory)
```

4.2. Dùng vòng lặp để xác định số lượng nguyên liệu cần đặt theo MOQ

- Logic xác định lượng nguyên liệu cần đặt được khái quát qua sơ đồ sau:
- Dùng vòng lặp để xử lý từng kỳ, trong đó, logic để xử lý từng kỳ gồm:

4.2.1 Xác định những ngày có trong 1 kỳ

```
# Xác định những ngày có trong từng kỳ
period_days = [(first_date_period + datetime.timedelta(days=day)) for day in range(age_of_inventory+1)]
# Xác định df_demand của từng kỳ
df_demand = df_daily_production_merged[df_daily_production_merged['job_start_date'].isin(period_days)]
```

Bước 4: Duyệt qua từng kỳ để tìm số lượng nguyên liệu cần đặt theo MOQ

4.2.2 Tính lũy kế lượng nguyên liệu cần của 1 nguyên liệu theo từng kỳ

```
# cumsum lượng nguyên liệu cần của 1 nguyên liệu theo từng kỳ  
df_demand['cumsum_raw_demand'] = df_demand.groupby(by=[df_demand['NGUYÊN LIỆU']])['cumsum_daily_demand_raw_qty'].cumsum()
```

4.2.3 Merge df_demand và df_supply để tìm ra lượng nguyên liệu thiếu

```
# Merge df_demand và df_supply  
df_merged = df_demand.merge(df_supply, how='left', on='NGUYÊN LIỆU')  
# Điền những giá trị NaN hoặc NaT do không có supply, đảm bảo logic supply không còn hàng  
df_merged['NGÀY VỀ'].fillna(pd.to_datetime('2000-1-1'), inplace=True)  
df_merged['SỐ LƯỢNG'].fillna(0, inplace=True)  
df_merged['HẠN SỬ DỤNG'].fillna(pd.to_datetime('2000-1-1'), inplace=True)
```

4.2.4 Xử lý những nguyên liệu không đủ supply

```
# Xử lý supply sau khi merge  
df_merged = process_supply(df_merged)  
# Xử lý những nguyên liệu không đủ supply  
df_MOQ_order, df_supply = process_df_insufficient(df_merged, df_MOQ_order, df_supply)
```

Bước 4: Duyệt qua từng kỳ để tìm số lượng nguyên liệu cần đặt theo MOQ

*Giải thích hàm process_supply(df_in):

```
def process_supply(df_in):
    # Nếu hết HẠN SỬ DỤNG thì SỐ LƯỢNG nguyên liệu supply sẽ bằng 0
    df_in.loc[df_in[df_in['job_start_date']] >= df_in['HẠN SỬ DỤNG']].index, 'SỐ LƯỢNG' = 0
    # Nếu hàng chưa về thì SỐ LƯỢNG nguyên liệu supply sẽ bằng 0
    df_in.loc[df_in[df_in['job_start_date']] < df_in['NGÀY VỀ']].index, 'SỐ LƯỢNG' = 0
    # Nếu hàng đã lưu kho quá số ngày tồn kho trung bình thì SỐ LƯỢNG nguyên liệu supply sẽ bằng 0
    df_in.loc[(df_in['job_start_date'] - df_in['NGÀY VỀ']) > pd.Timedelta(days=age_of_inventory))
              & (df_in['NGÀY VỀ'] != pd.to_datetime('2000-1-1'))].index, 'SỐ LƯỢNG' = 0

    # Cumsum supply theo từng ngày và theo từng nguyên liệu
    df_in['cumsum_raw_supply'] = df_in.groupby(['job_start_date', 'NGUYÊN LIỆU'])['SỐ LƯỢNG'].cumsum()

    return df_in
```

Đảm bảo logic để tính toán số lượng nguyên liệu thiếu theo HẠN SỬ DỤNG, ONGOING SUPPLY và SỐ NGÀY LƯU KHO TỐI ĐA

Bước 4: Duyệt qua từng kỳ để tìm số lượng nguyên liệu cần đặt theo MOQ

*Giải thích hàm process_df_insufficient(df_in, df_MOQ_order, df_supply): hàm trả về df_MOQ_order và df_supply được update để chạy trong kỳ tiếp theo

```
def process_df_insufficient(df_in, df_MOQ_order, df_supply):
    # Lấy ra giá trị cuối cùng theo từng nguyên liệu và từng ngày để xác định nguyên liệu bị thiếu supply
    df_insufficient_last = df_in.groupby(by=[df_in['job_start_date'], df_in['NGUYÊN LIỆU']],
                                         group_keys=False,
                                         sort=False).last()
    df_insufficient_last.reset_index(inplace=True)

    # Xác định những nguyên liệu bị thiếu
    df_insufficient_last = df_insufficient_last[df_insufficient_last['cumsum_raw_demand'] > df_insufficient_last['cumsum_raw_supply']]
    if len(df_insufficient_last) == 0:
        return df_MOQ_order, df_supply

    # Xác định lượng nguyên liệu bị thiếu theo từng ngày
    df_insufficient_last['insufficient_qty'] = df_insufficient_last['cumsum_raw_supply'] - df_insufficient_last['cumsum_raw_demand']
    df_insufficient_last.loc[df_insufficient_last['cumsum_raw_supply']==0].index, 'insufficient_qty'] = np.nan
    df_insufficient_last.loc[df_insufficient_last['cumsum_raw_supply']==0].index, 'cumsum_daily_demand_raw_qty'] = np.nan

    # Xác định tổng số lượng nguyên liệu bị thiếu cần phải đặt hàng
    df_insufficient_last = df_insufficient_last.groupby('NGUYÊN LIỆU').apply(process_insufficient_qty)

    # Tìm ngày đầu tiên bị thiếu hàng để đặt toàn bộ lượng hàng bị thiếu vào ngày đó
    df_insufficient_first = df_insufficient_last.groupby(by=[df_insufficient_last['NGUYÊN LIỆU'], df_insufficient_last['order_qty']],
                                                       group_keys=False,
                                                       sort=False).first()
    df_insufficient_first.reset_index(inplace=True)
    df_insufficient_first = df_insufficient_first[['NGUYÊN LIỆU', 'order_qty', 'job_start_date']]
```

Bước 4: Duyệt qua từng kỳ để tìm số lượng nguyên liệu cần đặt theo MOQ

*Giải thích hàm process_df_insufficient(df_in, df_MOQ_order, df_supply): hàm trả về df_MOQ_order và df_supply được update để chạy trong kỳ tiếp theo

```
def process_df_insufficient(df_in, df_MOQ_order, df_supply):
    # Lấy ra giá trị cuối cùng theo từng nguyên liệu và từng ngày để xác định nguyên liệu bị thiếu supply
    df_insufficient_last = df_in.groupby(by=[df_in['job_start_date'], df_in['NGUYÊN LIỆU']],
                                         group_keys=False,
                                         sort=False).last()
    df_insufficient_last.reset_index(inplace=True)

    # Xác định những nguyên liệu bị thiếu
    df_insufficient_last = df_insufficient_last[df_insufficient_last['cumsum_raw_demand'] > df_insufficient_last['cumsum_raw_supply']]
    if len(df_insufficient_last) == 0:
        return df_MOQ_order, df_supply

    # Xác định lượng nguyên liệu bị thiếu theo từng ngày
    df_insufficient_last['insufficient_qty'] = df_insufficient_last['cumsum_raw_supply'] - df_insufficient_last['cumsum_raw_demand']
    df_insufficient_last.loc[df_insufficient_last[df_insufficient_last['cumsum_raw_supply']==0].index, 'insufficient_qty'] = ^
    df_insufficient_last.loc[df_insufficient_last[df_insufficient_last['cumsum_raw_supply']==0].index, 'cumsum_daily_demand_raw_qty']

    # Xác định tổng số lượng nguyên liệu bị thiếu cần phải đặt hàng
    df_insufficient_last = df_insufficient_last.groupby('NGUYÊN LIỆU').apply(process_insufficient_qty)

    # Tìm ngày đầu tiên bị thiếu hàng để đặt toàn bộ lượng hàng bị thiếu vào ngày đó
    df_insufficient_first = df_insufficient_last.groupby(by=[df_insufficient_last['NGUYÊN LIỆU'], df_insufficient_last['order_qty']],
                                                       group_keys=False,
                                                       sort=False).first()
    df_insufficient_first.reset_index(inplace=True)
    df_insufficient_first = df_insufficient_first[['NGUYÊN LIỆU', 'order_qty', 'job_start_date']]
```

```
# Merge với df_ITEM_MASTER_RAW để tìm ra lượng đặt hàng theo MOQ
df_insufficient_first = df_insufficient_first.merge(df_ITEM_MASTER_RAW, on='NGUYÊN LIỆU', how='left')
df_insufficient_first = process_insufficient_first(df_insufficient_first)

# Update df_MOQ_order
df_MOQ_order = pd.concat([df_MOQ_order, df_insufficient_first[['NGUYÊN LIỆU', 'NHÀ CUNG CẤP', 'MOQ_order_qty', 'order_date']]], axis=0)

# Update df_supply
df_insufficient_first = df_insufficient_first[df_insufficient_first['SỐ LƯỢNG']!=0].rename(columns={'job_start_date':'NGÀY VỀ'})
df_supply = pd.concat([df_supply, df_insufficient_first[['NGUYÊN LIỆU', 'NGÀY VỀ', 'SỐ LƯỢNG', 'HẠN SỬ DỤNG']]], axis=0)

return df_MOQ_order, df_supply
```

Bước 4: Duyệt qua từng kỳ để tìm số lượng nguyên liệu cần đặt theo MOQ

4.2.5 Update df_supply để chạy trong kỳ tiếp theo

- Bỏ những nguyên liệu mà tới kỳ tiếp theo sẽ vi phạm quy định tồn kho tối đa trung bình

```
# Update df_supply
df_supply = df_supply[df_supply['NGÀY VỀ'].isin(period_days)]
df_supply = df_supply.sort_values(by=['NGUYÊN LIỆU', 'NGÀY VỀ'])
```

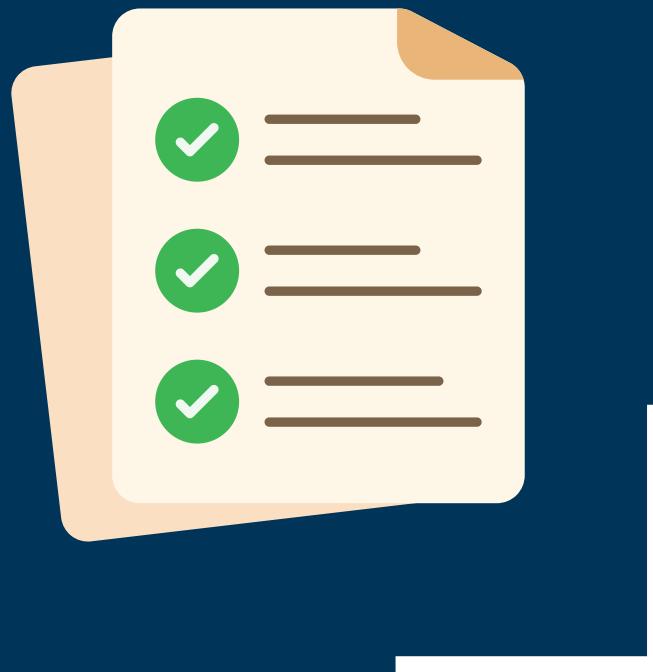
Bước 5: Xuất ra file excel chứa kết quả

```
# Pivot df_MOQ_order
df_MOQ_order_pivot = pd.pivot_table(df_MOQ_order,index=['NGUYÊN LIỆU', 'NHÀ CUNG CẤP'], columns=['order_date'], values=['MOQ_order_qty'])
df_MOQ_order_pivot.fillna("", inplace=True)
df_MOQ_order_pivot
```

		MOQ_order_qty													
		order_date	2023-01-14	2023-01-23	2023-01-25	2023-01-26	2023-01-27	2023-01-31	2023-02-02	2023-02-25	2023-03-02	2023-03-28	2023-04-02	2023-04-28	2023-05-03
NGUYÊN LIỆU	NHÀ CUNG CẤP														
BAO_BÌ_ĐÓNG_GÓI	CÔNG TY TNHH WOPACK				1921.290323				2070.709677		1970.580645		1920.086022		
BỘT_ĐỒ_TƯƠNG	CÔNG TY TNHH NARGI			6938.709677					6651.290323		7100.322581		7049.677419		
KHOAI	CÔNG TY TNHH NARGI				4666.359447				5919.354839		5119.354839		4900.645161		
MEN_VI_SINH	CÔNG TY VNIBIO	100.0				423.225806			517.677419		492.645161		480.021505		
NGÔ	CÔNG TY TNHH NARGI		20525.806452				22924.193548		22072.580645		21550.752688				



TEAM
HTM



TEAM
MHT

THANKS FOR READING

SC HACKFEST BY SCH