

VNA - PY THEO CÁCH CỦA BẠN

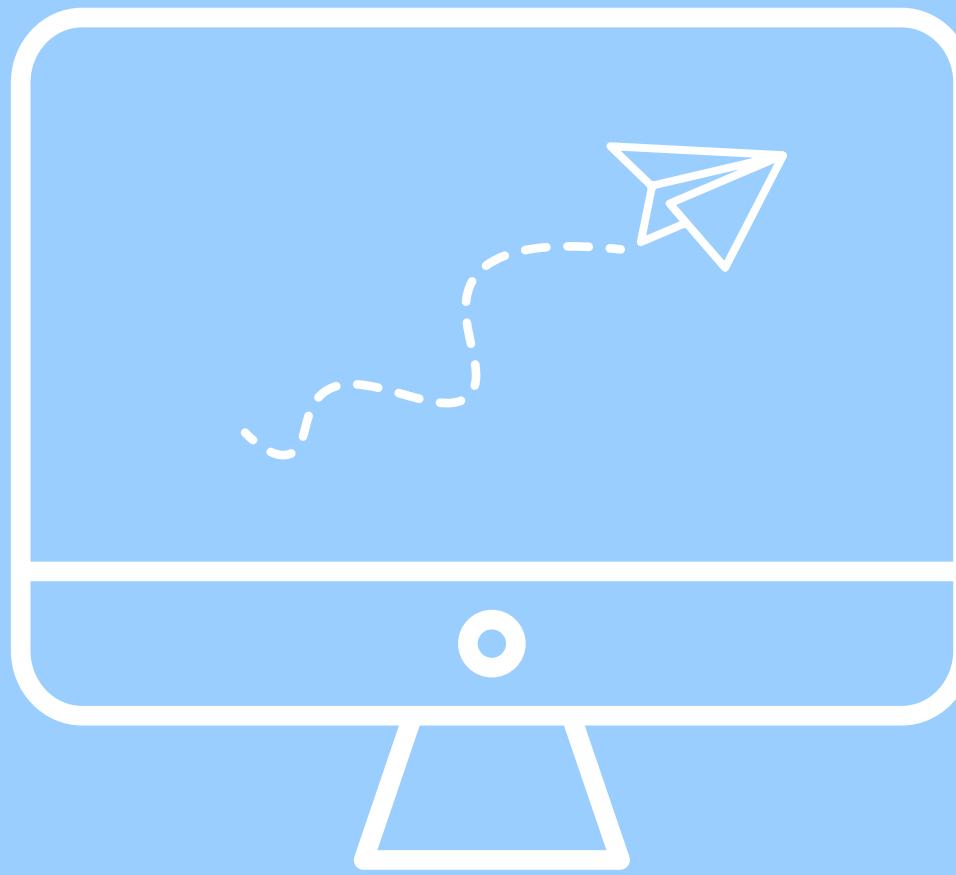
CASE STUDY: FESTORY

ROUND 2
SC HACKFEST 2022



GIỚI THIỆU THÀNH VIÊN

VNA - Py theo cách của bạn



Nguyễn My Na

- Sinh viên năm 2 Trường Đại học Ngoại thương Cơ sở II
- Chuyên ngành: Kinh tế đối ngoại

Dương Huỳnh Huyền Vi

- Sinh viên năm 2 Trường Đại học Ngoại thương Cơ sở II
- Chuyên ngành: Kinh tế đối ngoại

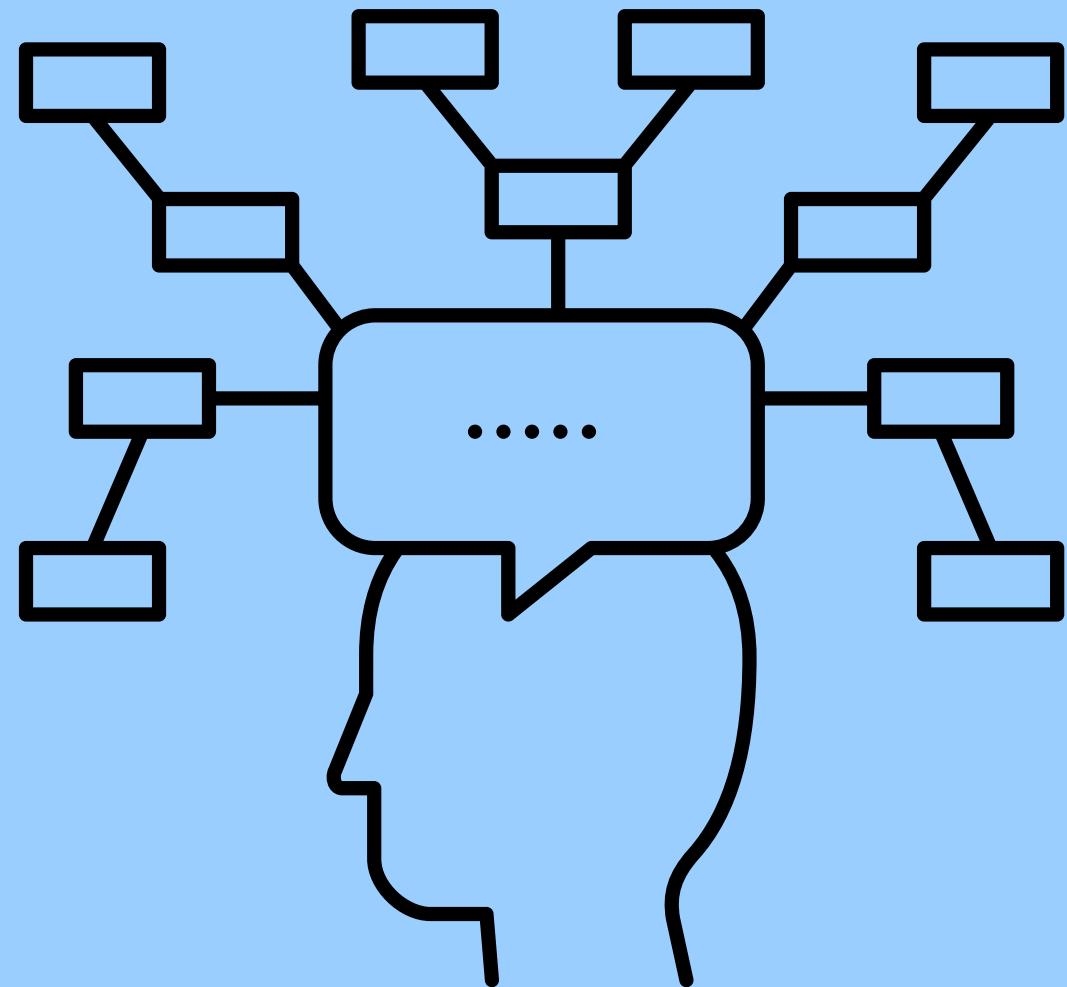
Phan Đình Bảo Anh

- Sinh viên năm 2 Trường Đại học Ngoại thương Cơ sở II
- Chuyên ngành: Kinh tế đối ngoại

VẤN ĐỀ ĐẶT RA

- Vấn đề 1: Đáp ứng đủ yêu cầu của khách hàng
- Vấn đề 2: Đúng quy tắc MOQ: đặt hàng từ mức tối thiểu mà nhà cung cấp yêu cầu
- Vấn đề 3: Số ngày tồn kho trung bình của các mặt hàng tối đa là 30 ngày.
- Vấn đề 4: Nguyên tắc FEFO

GIẢ THIẾT



- FESTORY chỉ sản xuất thành phẩm theo tháng
- Mặc định ngày xuất hiện nhu cầu của nguyên liệu của tháng là ngày đầu tháng đó
- Nhóm nhận thấy SHELF LIFE của các mặt hàng đều lớn hơn 30 ngày, do đó nếu nhập kho và sản xuất hết trong tháng thì sẽ không có hàng hết hạn tồn kho.
- Vì leadtime sản xuất của cả hai loại thành phẩm trong BẢNG 6_ITEM-FG đều bằng 0 nên những tính toán liên quan đến leadtime sản xuất sẽ được nhóm bỏ qua.
- Bao bì đóng gói không có hạn sử dụng nên Nhóm đặt là 30/01/2022 và SHELF LIFE lớn hơn 30 ngày.

BƯỚC 1

TÌM ĐƯỢC NHU CẦU CỦA
NGUYÊN LIỆU VÀ NGUỒN CUNG
HIỆN TẠI CỦA NGUYÊN LIỆU

TÍNH TOÁN NHU CẦU NGUYÊN LIỆU DỰA TRÊN Ý TƯỞNG:

Nhu cầu nguyên liệu = Nhu cầu thành phẩm * Số lượng nguyên liệu cần cho một tấn thành phẩm

- Đọc file excel Bảng 1_DEMAND

```
DEMAND=pd.read_excel('DATA_ROUND2_SCHACKFEST2022.xlsx',sheet_name='1_DEMAND')
```

DEMAND

- Đặt lại tên cột để thuận tiện cho việc tính toán sau này bằng cách dùng hàm rename và kiểu dữ liệu dictionary
- Dùng hàm melt để unpivot dataframe DEMAND, sử dụng hàm rename để đặt lại tên các cột
 - Kết quả:

	ITEM	DEMAND_MONTH	DEMAND_QTY
0	THỨC ĂN CHO HEO	2023-01	18
1	THỨC ĂN CHO CÁ	2023-01	16
2	THỨC ĂN CHO HEO	2023-02	22
3	THỨC ĂN CHO CÁ	2023-02	21
4	THỨC ĂN CHO HEO	2023-03	30
5	THỨC ĂN CHO CÁ	2023-03	22
6	THỨC ĂN CHO HEO	2023-04	25
7	THỨC ĂN CHO CÁ	2023-04	23
8	THỨC ĂN CHO HEO	2023-05	24
9	THỨC ĂN CHO CÁ	2023-05	23
10	THỨC ĂN CHO HEO	2023-06	27
11	THỨC ĂN CHO CÁ	2023-06	26

BƯỚC 1

TÌM ĐƯỢC NHU CẦU CỦA NGUYÊN LIỆU VÀ NGUỒN CUNG HIỆN TẠI CỦA NGUYÊN LIỆU

- Vì cuối tháng 12/2022 có lượng thành phẩm (lượng THỨC_ĂN_CHO_HEO và THỨC_ĂN_CHO_CÁ) còn tồn lại, do đó cần phải tính toán lại nhu cầu thành phẩm trong tháng 1.
- Dùng hàm str.contains để tìm ra Lượng Thành phẩm cần cho tháng 1 từ Dataframe DEMAND

ITEM	DEMAND_MONTH	DEMAND_QTY
0 THỨC_ĂN_CHO_HEO	2023-01	18
1 THỨC_ĂN_CHO_CÁ	2023-01	16

- Dùng lệnh drop để xóa Nhu cầu thành phẩm tháng 1 trong Dataframe DEMAND.
- Đọc file excel BẢNG 3_SUPPLY-SOH

SUPPLY_SOH=pd.read_excel('DATA_ROUND2_SCHACKFEST2022.xlsx',sheet_name='3_SUPPLY-SOH')

SUPPLY_SOH

- Để tạo sự đồng nhất với Bảng 4_SUPPLY-ONGOING sẽ được dùng về sau, tạo cột “NGÀY VỀ” mặc định là “31/12/2022”, đồng thời đổi tên cột “TỒN KHO CUỐI NGÀY 31/12/2022” thành “SỐ LƯỢNG”

▪ Kết quả:

ITEM	SỐ LƯỢNG	ĐƠN VỊ	HẠN SỬ DỤNG	NGÀY VỀ
0 KHOAI	10000	kg	2023-02-12	2022-12-31
1 SẴN	3000	kg	2023-02-27	2022-12-31
2 XƯƠNG_ĐỘNG_VẬT	2000	kg	2023-01-26	2022-12-31
3 BỘT_ĐỖ_TƯƠNG	9000	kg	2023-02-20	2022-12-31
4 MEN_VI_SINH	200	lít	2023-01-25	2022-12-31
5 NGÔ	12000	kg	2023-02-02	2022-12-31
6 THÓC	13000	kg	2023-03-03	2022-12-31
7 BAO_BÌ_ĐÓNG_GÓI	1220	cái	2023-01-30	2022-12-31
8 THỨC_ĂN_CHO_HEO	7000	kg	2023-01-27	2022-12-31
9 THỨC_ĂN_CHO_CÁ	8000	kg	2023-01-30	2022-12-31

BƯỚC 1

TÌM ĐƯỢC NHU CẦU CỦA
NGUYÊN LIỆU VÀ NGUỒN CUNG
HIỆN TẠI CỦA NGUYÊN LIỆU

- Đọc file excel BẢNG 3_SUPPLY-ONGOING

```
SUPPLY_ONGOING=pd.read_excel('DATA_ROUND2_SCHACKFEST2022.xlsx',sheet_name='4_SUPPLY-ONGOING')
```

SUPPLY_ONGOING

- Xóa cột "MÃ ĐƠN ĐẶT HÀNG" bằng lệnh del, đồng thời đặt lại tên cột "NGUYÊN LIỆU" thành "ITEM" để tiện cho việc ghép hai dataframe sau này.

- Kết quả:

	ITEM	NGÀY VỀ	SỐ LƯỢNG	ĐƠN VỊ	HẠN SỬ DỤNG
0	KHOAI	2023-01-03	1800	kg	2023-04-13
1	NGÔ	2023-01-05	1000	kg	2023-04-15
2	SẮN	2023-01-12	1000	kg	2023-04-22

- Thực hiện ghép hai Dataframe SUPPLY_SOH và SUPPLY_ONGOING bằng lệnh concat. Tên file ghép là SUPPLY.
- Chèn thêm cột "NGÀY HẾT HẠN TỒN KHO" để tính toán yêu cầu không được để hàng quá 30 ngày trong kho của Ban Giám đốc.
- NGÀY HẾT HẠN TỒN KHO = NGÀY VỀ + 30 (ngày)

- Kết quả:

	ITEM	SỐ LƯỢNG	ĐƠN VỊ	HẠN SỬ DỤNG	NGÀY VỀ	NGÀY HẾT HẠN TỒN KHO
0	KHOAI	10000	kg	2023-02-12	2022-12-31	2023-01-30
1	SẮN	3000	kg	2023-02-27	2022-12-31	2023-01-30
2	XƯƠNG_ĐỘNG_VẬT	2000	kg	2023-01-26	2022-12-31	2023-01-30
3	BỘT_ĐỖ_TƯƠNG	9000	kg	2023-02-20	2022-12-31	2023-01-30
4	MEN_VI_SINH	200	lít	2023-01-25	2022-12-31	2023-01-30
5	NGÔ	12000	kg	2023-02-02	2022-12-31	2023-01-30
6	THÓC	13000	kg	2023-03-03	2022-12-31	2023-01-30
7	BAO_BÌ_ĐÓNG_GÓI	1220	cái	2023-01-30	2022-12-31	2023-01-30
8	THỨC_ĂN_CHO_HEO	7000	kg	2023-01-27	2022-12-31	2023-01-30
9	THỨC_ĂN_CHO_CÁ	8000	kg	2023-01-30	2022-12-31	2023-01-30
0	KHOAI	1800	kg	2023-04-13	2023-01-03	2023-02-02
1	NGÔ	1000	kg	2023-04-15	2023-01-05	2023-02-04
2	SẮN	1000	kg	2023-04-22	2023-01-12	2023-02-11

BƯỚC 1

TÌM ĐƯỢC NHU CẦU CỦA
NGUYÊN LIỆU VÀ NGUỒN CUNG
HIỆN TẠI CỦA NGUYÊN LIỆU

- Tạo Dataframe các sản phẩm thừa từ tháng 12/2023:

```
SẢN_PHẨM_THỪA=SUPPLY[SUPPLY['ITEM'].str.contains('THỨC ĂN_')]  
SẢN_PHẨM_THỪA=SẢN_PHẨM_THỪA.rename(columns={'SỐ LƯỢNG':'SỐ LƯỢNG ĐANG TỒN'})  
SẢN_PHẨM_THỪA
```

	ITEM	SỐ LƯỢNG ĐANG TỒN	ĐƠN VỊ	HẠN SỬ DỤNG	NGÀY VỀ	NGÀY HẾT HẠN TỒN KHO
8	THỨC ĂN CHO HEO	7000	kg	2023-01-27	2022-12-31	2023-01-30
9	THỨC ĂN CHO CÁ	8000	kg	2023-01-30	2022-12-31	2023-01-30

- Thực hiện đổi đơn vị từ kg sang tấn

	ITEM	SỐ LƯỢNG ĐANG TỒN	ĐƠN VỊ	HẠN SỬ DỤNG	NGÀY VỀ	NGÀY HẾT HẠN TỒN KHO
8	THỨC ĂN CHO HEO	7.0	kg	2023-01-27	2022-12-31	2023-01-30
9	THỨC ĂN CHO CÁ	8.0	kg	2023-01-30	2022-12-31	2023-01-30

- Kết hợp với Dataframe DEMAND_THANG1, tính toán lại số lượng thành phẩm cần trong tháng 1 theo công thức:

SỐ LƯỢNG = DEMAND_QTY – SỐ LƯỢNG ĐANG TỒN

- Kết quả

```
DEMAND_NEW=SẢN_PHẨM_THỪA.merge(DEMAND_THANG1,how='left',on='ITEM')  
DEMAND_NEW
```

	ITEM	SỐ LƯỢNG ĐANG TỒN	ĐƠN VỊ	HẠN SỬ DỤNG	NGÀY VỀ	NGÀY HẾT HẠN TỒN KHO	DEMAND_MONTH	DEMAND_QTY
0	THỨC ĂN CHO HEO	7.0	kg	2023-01-27	2022-12-31	2023-01-30	2023-01	18
1	THỨC ĂN CHO CÁ	8.0	kg	2023-01-30	2022-12-31	2023-01-30	2023-01	16

BƯỚC 1

TÌM ĐƯỢC NHU CẦU CỦA
NGUYÊN LIỆU VÀ NGUỒN CUNG
HIỆN TẠI CỦA NGUYÊN LIỆU

- Thực hiện đổi tên các cột cho phù hợp và merge DEMAND_NEW với DEMAND, gọi chung là DEMAND_NEW để xuất ra nhu cầu thành phẩm cuối cùng:

```
DEMAND_NEW=DEMAND_NEW[['ITEM', 'DEMAND_MONTH', 'DEMAND_QTY']]  
DEMAND_NEW=pd.concat([DEMAND_NEW,DEMAND],axis=0)  
DEMAND_NEW
```

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

- Đọc file SỐ LƯỢNG NGUYÊN LIỆU CẦN CHO MỘT TẤN THÀNH PHẨM:

```
BOM=pd.read_excel('DATA_ROUND2_SCHACKFEST2022.xlsx',sheet_name='2_BOM')  
  
BOM=BOM.rename(columns={'THÀNH PHẨM':'ITEM'})  
BOM
```

BƯỚC 1

TÌM ĐƯỢC NHU CẦU CỦA
NGUYÊN LIỆU VÀ NGUỒN CUNG
HIỆN TẠI CỦA NGUYÊN LIỆU

- Tạo dataframe DEMAND_MERGE: Dùng lệnh merge để kết hợp hai BẢNG BOM và DEMAND_NEW, merge kiểu inner và dựa trên “ITEM”, đổi tên các cột cho phù hợp
 - Kết quả:

	THÀNH PHẨM	DEMAND_MONTH	DEMAND_QTY	ITEM	SỐ LƯỢNG NGUYÊN LIỆU \n(TRÊN 1 TẤN THÀNH PHẨM)	ĐƠN VỊ CỦA NGUYÊN LIỆU
0	THỨC ĂN CHO HEO	2023-01	11.0	NGÔ	400	kg
1	THỨC ĂN CHO HEO	2023-01	11.0	KHOAI	200	kg
2	THỨC ĂN CHO HEO	2023-01	11.0	SẮN	200	kg
3	THỨC ĂN CHO HEO	2023-01	11.0	XƯƠNG ĐỘNG VẬT	200	kg
4	THỨC ĂN CHO HEO	2023-01	11.0	MEN VI SINH	10	lít
...
67	THỨC ĂN CHO CÁ	2023-06	26.0	BỘT ĐỒ TƯƠNG	300	kg
68	THỨC ĂN CHO CÁ	2023-06	26.0	THÓC	100	kg
69	THỨC ĂN CHO CÁ	2023-06	26.0	XƯƠNG ĐỘNG VẬT	100	kg
70	THỨC ĂN CHO CÁ	2023-06	26.0	MEN VI SINH	10	lít
71	THỨC ĂN CHO CÁ	2023-06	26.0	BAO BÌ ĐÓNG GÓI	40	cái

- Tính toán tổng lượng cần cho từng nguyên liệu cụ thể cho từng tháng:
DEMAND_MERGE['NGLIEU_DEMAND_QTY']=DEMAND_MERGE['DEMAND_QTY']*DEMAND_MERGE['SỐ LƯỢNG NGUYÊN LIỆU \n(TRÊN 1 TẤN THÀNH PHẨM)']
 - Kết quả:

	THÀNH PHẨM	DEMAND_MONTH	DEMAND_QTY	ITEM	SỐ LƯỢNG NGUYÊN LIỆU \n(TRÊN 1 TẤN THÀNH PHẨM)	ĐƠN VỊ CỦA NGUYÊN LIỆU	NGLIEU_DEMAND_QTY
	THỨC ĂN CHO HEO	2023-01	11.0	NGÔ	400	kg	4400.0
	THỨC ĂN CHO HEO	2023-01	11.0	KHOAI	200	kg	2200.0
	THỨC ĂN CHO HEO	2023-01	11.0	SẮN	200	kg	2200.0
	THỨC ĂN CHO HEO	2023-01	11.0	XƯƠNG ĐỘNG VẬT	200	kg	2200.0
	THỨC ĂN CHO HEO	2023-01	11.0	MEN VI SINH	10	lít	110.0
	THỨC ĂN CHO HEO	2023-01	11.0	BAO BÌ ĐÓNG GÓI	40	cái	440.0
	THỨC ĂN CHO HEO	2023-02	22.0	NGÔ	400	kg	8800.0

BƯỚC 1

TÌM ĐƯỢC NHU CẦU CỦA
NGUYÊN LIỆU VÀ NGUỒN CUNG
HIỆN TẠI CỦA NGUYÊN LIỆU

- Tạo Dataframe DEMAND_FINAL: dùng groupby để xử lý DEMAND_MERGE, group theo ITEM, DEMAND_MONTH và tính sum của NGLIEU_DEMAND_QTY
 - Kết quả:

	ITEM	DEMAND_MONTH	NGLIEU_DEMAND_QTY
0	BAO_BÌ_ĐÓNG_GÓI	2023-01	760
1	BAO_BÌ_ĐÓNG_GÓI	2023-02	1720
2	BAO_BÌ_ĐÓNG_GÓI	2023-03	2080
3	BAO_BÌ_ĐÓNG_GÓI	2023-04	1920
4	BAO_BÌ_ĐÓNG_GÓI	2023-05	1880
5	BAO_BÌ_ĐÓNG_GÓI	2023-06	2120
6	BỘT_ĐỖ_TƯƠNG	2023-01	2400
7	BỘT_ĐỖ_TƯƠNG	2023-02	6300
8	BỘT_ĐỖ_TƯƠNG	2023-03	6600
9	BỘT_ĐỖ_TƯƠNG	2023-04	6900
10	BỘT_ĐỖ_TƯƠNG	2023-05	6900
11	BỘT_ĐỖ_TƯƠNG	2023-06	7800
12	KHOAI	2023-01	2200
13	KHOAI	2023-02	4400

TÍNH TOÁN LƯỢNG CUNG NGUYÊN LIỆU HIỆN TẠI (BAO GỒM LƯỢNG HÀNG SOH VÀ ONGOING)

- Từ DataFrame SUPPLY đã tạo từ trước, sort theo ITEM và NGÀY HẾT HẠN TỒN KHO.

BƯỚC 2

TẠO FILE TỔNG HỢP GIỮA DEMAND_FINAL VÀ SUPPLY

- Tạo Dataframe DF_MERGE: merge hai Dataframe DEMAND_FINAL và SUPPLY theo tên ITEM (nguyên liệu) và theo tháng (tháng xuất hiện nhu cầu và tháng hết hạn sử dụng của nguyên liệu hiện có: bao gồm SOH và ONGOING)
 - Kết quả:

	ITEM	DEMAND_MONTH	NGLIEU_DEMAND_QTY	MONTH	SỐ LƯỢNG	ĐƠN VỊ	HẠN SỬ DỤNG	NGÀY VỀ	NGÀY HẾT HẠN TỒN KHO
0	BAO_BÌ_ĐÓNG_GÓI	2023-01-01		760	1	1220.0	cái	2023-01-30 00:00:00	2022-12-31 00:00:00 2023-01-30 00:00:00
1	BAO_BÌ_ĐÓNG_GÓI	2023-02-01		1720	2	0.0	0	0	0
2	BAO_BÌ_ĐÓNG_GÓI	2023-03-01		2080	3	0.0	0	0	0
3	BAO_BÌ_ĐÓNG_GÓI	2023-04-01		1920	4	0.0	0	0	0
4	BAO_BÌ_ĐÓNG_GÓI	2023-05-01		1880	5	0.0	0	0	0

- Chú thích: hai cột cần quan tâm nhiều nhất là NGLIEU_DEMAND_QTY (lượng nguyên liệu cần) và SỐ LƯỢNG (lượng nguyên liệu có trong kho và sắp về đến kho).
- Dựa vào hai cột này, xác định lượng nguyên liệu thừa/thiếu của mỗi tháng theo công thức:
$$\text{THỪA/THIẾU} = \text{SỐ LƯỢNG} - \text{NGLIEU_DEMAND_QTY}$$
 - Nếu kết quả là số âm tức là nglieu bị thiếu
 - Nếu kết quả dương là nglieu sẽ bị dư so với lượng cần của tháng đó
- Dùng groupby (group theo ITEM) và cumsum để cumsum lượng thừa hay thiếu ấy.
 - Kết quả

	ITEM	DEMAND_MONTH	NGLIEU_DEMAND_QTY	SỐ LƯỢNG	HẠN SỬ DỤNG	THỪA/THIẾU	CUMSUM_THỪA/THIẾU
1	BAO_BÌ_ĐÓNG_GÓI	2023-01-01		760	1220 01-30 00:00:00	460	460
	BAO_BÌ_ĐÓNG_GÓI	2023-02-01		1720	0	-1720	-1260

BƯỚC 2

TẠO FILE TỔNG HỢP GIỮA DEMAND_FINAL VÀ SUPPLY

- Đọc file excel BẢNG 5 _ ITEM-MASTER

```
ITEM_MASTER=pd.read_excel('DATA_ROUND2_SCHACKFEST2022.xlsx',sheet_name='5_ITEM MASTER - RAW')
del ITEM_MASTER['Loại']
del ITEM_MASTER['Đơn vị']
ITEM_MASTER
```

- Dùng hàm merge để kết hợp hai Dataframe ITEM-MASTER và DF_MERGE lại với nhau, tạo thành file mới là DF_MERGE2. Kể từ đây, các thao tác tính toán sẽ sử dụng Dataframe DF_MERGE2.
 - Kết quả:

	ITEM	DEMAND_MONTH	NGLIEU_DEMAND_QTY	SỐ LƯỢNG	HẠN SỬ DỤNG	THỪA/THIẾU	CUMSUM_THỪA/THIẾU	NHÀ CUNG CẤP	LEADTIME ĐẶT HÀNG (NGÀY)	ĐƯỢC ƯỚC TÍNH KẾ TỪ NGÀY NHẬP KHO	SHELF LIFE	
											MOQ	(NGÀY)
0	BAO_BÌ_ĐÓNG_GÓI	2023-01-01	760	1220	2023-01-30 00:00:00	460	460	CÔNG TY TNHH WOPACK	7	500	NaN	
1	BAO_BÌ_ĐÓNG_GÓI	2023-02-01	1720	0	0	-1720	-1260	CÔNG TY TNHH WOPACK	7	500	NaN	
2	BAO_BÌ_ĐÓNG_GÓI	2023-03-01	2080	0	0	-2080	-3340	CÔNG TY TNHH WOPACK	7	500	NaN	
3	BAO_BÌ_ĐÓNG_GÓI	2023-04-01	1020	0	0	-1020	5260	CÔNG TY	7	500	NaN	

BƯỚC 3

TAO CÁC DATAFRAME GỒM NHỮNG MẶT HÀNG CẦN ĐẶT THEO TỪNG ĐƠT

- Tạo DataFrame DAT_HANG: đây là Dataframe thể hiện các ITEM cần đặt, đó là những dòng có CUMSUM_THỪA/THIẾU <0 được lọc từ Dataframe DF_MERGE2.
 - Kết quả:

ITEM	DEMAND_MONTH	NGLIEU_DEMAND_QTY	SỐ LƯỢNG	HẠN SỬ DỤNG	THỪA/THIẾU	CUMSUM_THỪA/THIẾU	NHÀ CUNG CẤP	LEADTIME ĐẶT HÀNG (NGÀY)	MOQ	SHELF LIFE
										ĐƯỢC UỐC
										TÍNH KẾ TỪ NGÀY NHẬP KHO (NGÀY)
1 BAO_BÌ_ĐÓNG_GÓI	2023-02-01	1720	0	0	-1720	-1260	CÔNG TY TNHH WOPACK	7	500	NaN
2 BAO_BÌ_ĐÓNG_GÓI	2023-03-01	2080	0	0	-2080	-3340	CÔNG TY TNHH WOPACK	7	500	NaN
3 BAO_BÌ_ĐÓNG_GÓI	2023-04-01	1920	0	0	-1920	-5260	CÔNG TY TNHH WOPACK	7	500	NaN
4 BAO BÌ ĐÓNG GÓI	2023-05-01	1880	0	0	-1880	-7140	CÔNG TY	7	500	NaN

- Tạo Dataframe DAT_HANG1: đây là dataframe chứa các ITEM cần đặt trong đợt hết hàng đầu tiên của chúng.

- Dùng hàm groupby (theo ITEM) và first để chọn ra những dòng ITEM có CUMSUM_THỪA/THIẾU âm đầu tiên của mỗi nguyên liệu.

- Kết quả:

ITEM	DEMAND_MONTH	NGLIEU_DEMAND_QTY	SỐ LƯỢNG	HẠN SỬ DỤNG	THỪA/THIẾU	CUMSUM_THỪA/THIẾU	NHÀ CUNG CẤP	LEADTIME ĐẶT HÀNG (NGÀY)	MOQ	SHELF LIFE
										ĐƯỢC UỐC
										TÍNH KẾ TỪ NGÀY NHẬP KHO (NGÀY)
BAO_BÌ_ĐÓNG_GÓI	2023-02-01	1720	0	0	-1720	-1260	CÔNG TY TNHH WOPACK	7	500	NaN
BỘT ĐỔ TƯƠNG	2023-01-01	2400	0	0	-2400	-2400	CÔNG TY TNHH NARGI	7	1000	100.0
KHOAI	2023-01-01	2200	0	0	-2200	-2200	CÔNG TY TNHH NARGI	7	1000	100.0

TẠO CÁC DATAFRAME GỒM NHỮNG MẶT HÀNG CẦN ĐẶT THEO TỪNG ĐỢT

BƯỚC 3

- Tạo cột NGÀY ĐẶT theo công thức:
 - NGÀY ĐẶT = NGÀY THIẾU (DEMAND_MONTH) – LEADTIME ĐẶT HÀNG – LEADTIME SẢN XUẤT
 - Như đã nói lúc đầu, Leadtime sản xuất =0 nên sẽ được nhóm bỏ qua.
 - Kết quả:

ITEM	DEMAND_MONTH	NGLIEU_DEMAND_QTY	SỐ LƯỢNG	HẠN SỬ DỤNG	THẢA/THIẾU	CURSUM_THẢA/THIẾU	NHÀ CUNG CẤP	LEADTIME ĐẶT HÀNG (NGÀY)	MOQ	SHELF LIFE ĐƯỢC UỐC TÍNH KỂ TỪ NGÀY NHẬP KHO (NGÀY)	NGÀY ĐẶT
BAO_BÌ_DÔNG_GÓI	2023-02-01	1720	0	0	-1720	-1260	CÔNG TY TNHH WOPACK	7	500	NaN	2023-01-25
BỘT_DÔ_TƯƠNG	2023-01-01	2400	0	0	-2400	-2400	CÔNG TY TNHH NARGI	7	1000	100.0	2022-12-25
KHOAI	2023-01-01	2200	0	0	-2200	-2200	CÔNG TY TNHH NARGI	7	1000	100.0	2022-12-25
MEN_VỊ_SINH	2023-02-01	430	0	0	-430	-420	CÔNG TY VNIBIO	2	100	30.0	2023-01-25
NGÔ	2023-01-01	8400	0	0	-8400	-8400	CÔNG TY TNHH NARGI	7	3000	100.0	2022-12-25
SÁN	2023-01-01	2200	0	0	-2200	-2200	CÔNG TY TNHH NARGI	7	1000	100.0	2022-12-25
THÓC	2023-01-01	800	0	0	-800	-800	CÔNG TY TNHH NARGI	7	1000	100.0	2022-12-25
XƯƠNG_DÔNG_VẬT	2023-01-01	3000	2000	2023-01-26 00:00:00	-1000	-1000	CÔNG TY CP YFOOD	7	1000	30.0	2022-12-25

- Tạo cột LƯỢNG ĐẶT:
 - Để đảm bảo đúng quy tắc MOQ với nhà cung cấp, ta phải chia ra hai trường hợp:
 - Nếu lượng thiếu > lượng MOQ thì lượng đặt là lượng thiếu
 - Nếu lượng thiếu < lượng MOQ thì lượng đặt là MOQ.
 - Dựa trên tư duy đó, Nhóm tiến hành tách Dataframe DAT_HANG1 thành hai Dataframe nhỏ là dat_bang_thieu1 và dat_hon_thieu1 bằng cách lọc dữ liệu theo tư duy trên, sau đó ghép lại thành Dataframe DAT_HANG1 cuối cùng.
 - Kết quả:

ITEM	DEMAND_MONTH	NGLIEU_DEMAND_QTY	SỐ LƯỢNG	HẠN SỬ DỤNG	THẢA/THIẾU	CURSUM_THẢA/THIẾU	NHÀ CUNG CẤP	LEADTIME ĐẶT HÀNG (NGÀY)	MOQ	SHELF LIFE ĐƯỢC UỐC TÍNH KỂ TỪ NGÀY NHẬP KHO (NGÀY)	NGÀY ĐẶT	LƯỢNG ĐẶT
BAO_BÌ_DÔNG_GÓI	2023-02-01	1720	0	0	-1720	-1260	CÔNG TY TNHH WOPACK	7	500	NaN	2023-01-25	1260
BỘT_DÔ_TƯƠNG	2023-01-01	2400	0	0	-2400	-2400	CÔNG TY TNHH NARGI	7	1000	100.0	2022-12-25	2400
KHOAI	2023-01-01	2200	0	0	-2200	-2200	CÔNG TY TNHH NARGI	7	1000	100.0	2022-12-25	2200
MEN_VỊ_SINH	2023-02-01	430	0	0	-430	-420	CÔNG TY VNIBIO	2	100	30.0	2023-01-25	420
NGÔ	2023-01-01	8400	0	0	-8400	-8400	CÔNG TY TNHH NARGI	7	3000	100.0	2022-12-25	8400

BƯỚC 3

TAO CÁC DATAFRAME GỒM NHỮNG MẶT HÀNG CẦN ĐẶT THEO TỪNG ĐỢT

- Tạo Dataframe DAT_HANG2: là những ITEM sẽ bị thiếu lần thứ hai trong kỳ
 - Khi đã có lượng hàng đã đặt lần 1 (DAT_HANG1), merge với DF_MERGE2 để làm mới lượng nguyên liệu sẽ được cung cấp, vì lúc này đã có thêm lượng cung cấp.
 - Kết quả:

The screenshot shows a Jupyter Notebook cell with the following code:

```
DF_MERGE2=DF_MERGE2.merge(DAT_HANG1,how='left',on=['ITEM','DEMAND_MONTH','INGLIEU_DEMAND_QTY','SỐ LƯỢNG','HẠN SỬ DỤNG','THIỀU/THIẾU','CUSUM_THIỀU/THIẾU','Nhà cung cấp','LEADTIME ĐẶT HÀNG (NGÀY)','POQ','SHELF LIFE Được ước tính kể từ ngày nhập kho (ngày)'])  
DF_MERGE2.head()
```

Below the code, the resulting DataFrame DF_MERGE2 is displayed:

	ITEM	DEMAND_MONTH	INGLIEU_DEMAND_QTY	SỐ LƯỢNG	HẠN SỬ DỤNG	THIỀU/THIẾU	CUSUM_THIỀU/THIẾU	NHÀ CUNG CẤP	LEADTIME ĐẶT HÀNG (NGÀY)	POQ	SHELF LIFE Được ước tính kể từ ngày nhập kho (ngày)	NGÀY ĐẶT	LƯỢNG ĐẶT
0	BAO_BÌ_DÓNG_GÓI	2023-01-01	760	1220	2023-01-30 00:00:00	460	460	CÔNG TY TNHH WOPACK	7	500		NaN	NaN
1	BAO_BÌ_DÓNG_GÓI	2023-02-01	1720	0	0	-1720	-1260	CÔNG TY TNHH WOPACK	7	500		NaN	2023-01-25 1260.0
2	BAO_BÌ_DÓNG_GÓI	2023-03-01	2080	0	0	-2080	-3340	CÔNG TY TNHH WOPACK	7	500		NaN	NaN
3	BAO_BÌ_DÓNG_GÓI	2023-04-01	1920	0	0	-1920	-5260	CÔNG TY TNHH WOPACK	7	500		NaN	NaN
4	BAO_BÌ_DÓNG_GÓI	2023-05-01	1880	0	0	-1880	-7140	CÔNG TY TNHH WOPACK	7	500		NaN	NaN

- Tính toán lại cột “SỐ LƯỢNG” (lượng cung cấp mới), cột THIỀU/THIẾU và cột CUSUM_THIỀU/THIẾU.
- Làm tương tự như lúc tạo DAT_HANG1, ta được DAT_HANG2

The screenshot shows a Jupyter Notebook cell with the same code as the previous one:

```
DF_MERGE2=DF_MERGE2.merge(DAT_HANG1,how='left',on=['ITEM','DEMAND_MONTH','INGLIEU_DEMAND_QTY','SỐ LƯỢNG','HẠN SỬ DỤNG','THIỀU/THIẾU','CUSUM_THIỀU/THIẾU','Nhà cung cấp','LEADTIME ĐẶT HÀNG (NGÀY)','POQ','SHELF LIFE Được ước tính kể từ ngày nhập kho (ngày)'])  
DF_MERGE2.head()
```

Below the code, the resulting DataFrame DF_MERGE2 is displayed:

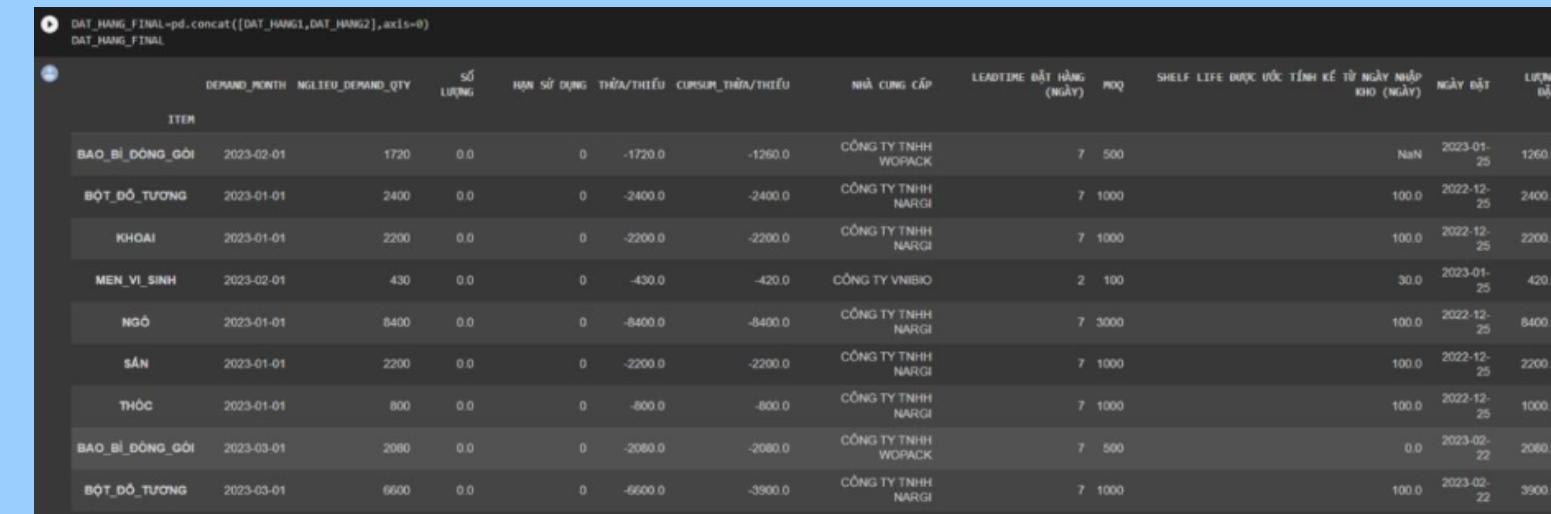
	ITEM	DEMAND_MONTH	INGLIEU_DEMAND_QTY	SỐ LƯỢNG	HẠN SỬ DỤNG	THIỀU/THIẾU	CUSUM_THIỀU/THIẾU	NHÀ CUNG CẤP	LEADTIME ĐẶT HÀNG (NGÀY)	POQ	SHELF LIFE Được ước tính kể từ ngày nhập kho (ngày)	NGÀY ĐẶT	LƯỢNG ĐẶT
0	BAO_BÌ_DÓNG_GÓI	2023-01-01	760	1220	2023-01-30 00:00:00	460	460	CÔNG TY TNHH WOPACK	7	500		NaN	NaN
1	BAO_BÌ_DÓNG_GÓI	2023-02-01	1720	0	0	-1720	-1260	CÔNG TY TNHH WOPACK	7	500		NaN	2023-01-25 1260.0
2	BAO_BÌ_DÓNG_GÓI	2023-03-01	2080	0	0	-2080	-3340	CÔNG TY TNHH WOPACK	7	500		NaN	NaN
3	BAO_BÌ_DÓNG_GÓI	2023-04-01	1920	0	0	-1920	-5260	CÔNG TY TNHH WOPACK	7	500		NaN	NaN
4	BAO_BÌ_DÓNG_GÓI	2023-05-01	1880	0	0	-1880	-7140	CÔNG TY TNHH WOPACK	7	500		NaN	NaN

- Tương tự như vậy cho những lần đặt hàng tiếp theo

BƯỚC 4

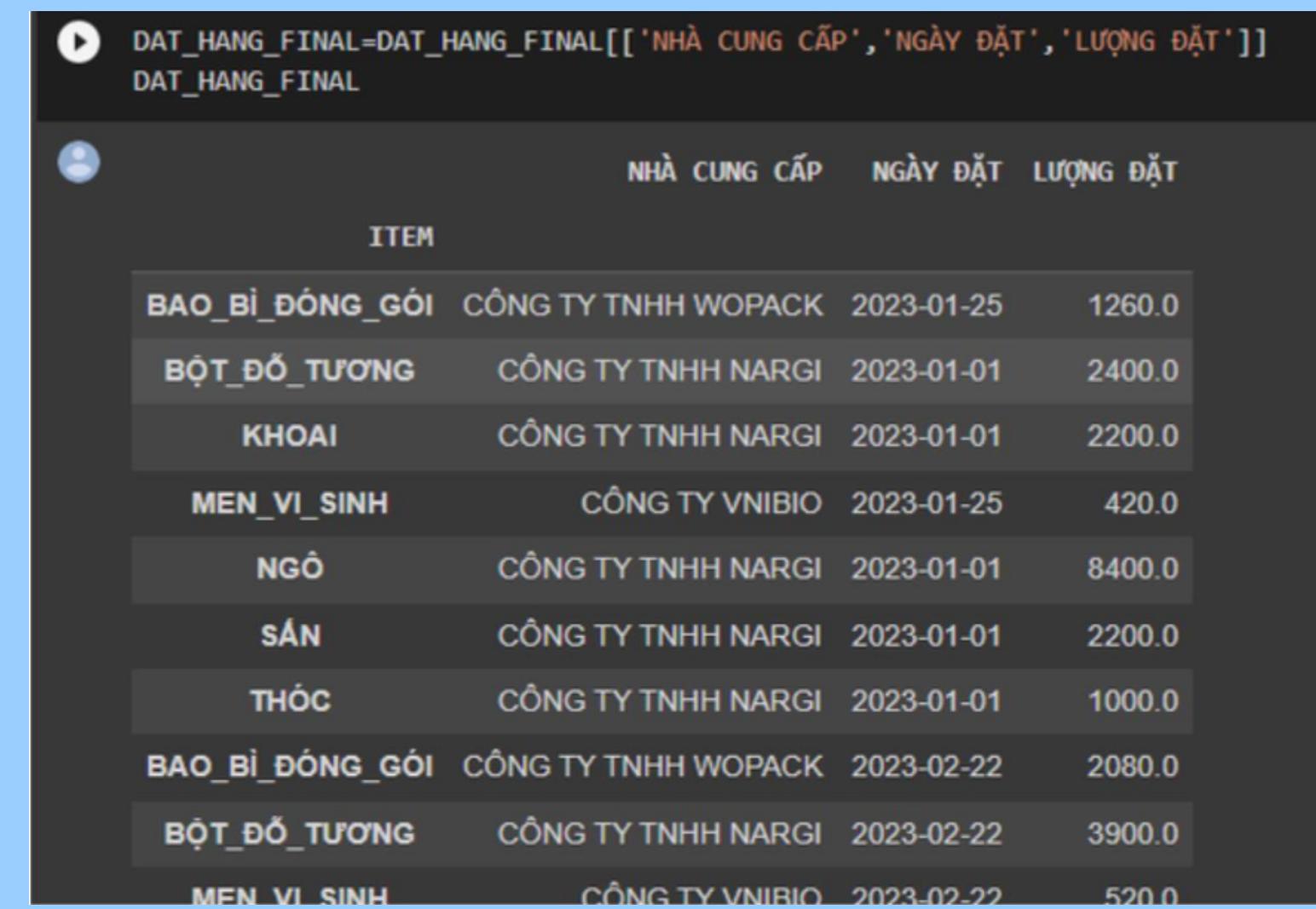
TỔNG HỢP CÁC ĐƠN ĐẶT HÀNG

- Dùng hàm concat để ghép các file DAT_HANG1, DAT_HANG2,... thành DAT_HANG_FINAL



ITEM	DEMAND_MONTH	NGLIEU_DEMAND_QTY	SỐ LƯỢNG	HẠN SỬ DỤNG THẢO/THIẾU	CURSUR_THẢO/THIẾU	NHÀ CUNG CẤP	LEADTIME ĐẶT HÀNG (NGÀY)	MỌQ	SHELF LIFE ĐƯỢC ƯỚC TÍNH KỂ TỪ NGÀY NHẬP KHÓ (NGÀY)	NGÀY ĐẶT	LƯỢNG ĐẶT
BAO_BÌ_ĐÓNG_GÓI	2023-02-01	1720	0.0	0	-1720.0	-1260.0	CÔNG TY TNHH WOPACK	7	500	NaN	2023-01-25 1260.0
BỘT_ĐỒ_TƯƠNG	2023-01-01	2400	0.0	0	-2400.0	-2400.0	CÔNG TY TNHH NARGI	7	1000	100.0	2022-12-25 2400.0
KHOAI	2023-01-01	2200	0.0	0	-2200.0	-2200.0	CÔNG TY TNHH NARGI	7	1000	100.0	2022-12-25 2200.0
MEN_VI_SINH	2023-02-01	430	0.0	0	-430.0	-420.0	CÔNG TY VNIBIO	2	100	30.0	2023-01-25 420.0
NGÔ	2023-01-01	8400	0.0	0	-8400.0	-8400.0	CÔNG TY TNHH NARGI	7	3000	100.0	2022-12-25 8400.0
SÁN	2023-01-01	2200	0.0	0	-2200.0	-2200.0	CÔNG TY TNHH NARGI	7	1000	100.0	2022-12-25 2200.0
THÓC	2023-01-01	800	0.0	0	-800.0	-800.0	CÔNG TY TNHH NARGI	7	1000	100.0	2022-12-25 1000.0
BAO_BÌ_ĐÓNG_GÓI	2023-03-01	2080	0.0	0	-2080.0	-2080.0	CÔNG TY TNHH WOPACK	7	500	0.0	2023-02-22 2080.0
BỘT_ĐỒ_TƯƠNG	2023-03-01	6600	0.0	0	-6600.0	-3900.0	CÔNG TY TNHH NARGI	7	1000	100.0	2023-02-22 3900.0

- Thay thế các ngày đặt hàng 25/12/2022 thành ngày 01/01/2023 bằng hàm Replace.
- Chọn xuất ra các cột ‘NHÀ CUNG CẤP’, ‘NGÀY ĐẶT’ và ‘LƯỢNG ĐẶT’, ta sẽ được Dataframe DAT_HANG_FINAL:



ITEM	NHÀ CUNG CẤP	NGÀY ĐẶT	LƯỢNG ĐẶT
BAO_BÌ_ĐÓNG_GÓI	CÔNG TY TNHH WOPACK	2023-01-25	1260.0
BỘT_ĐỒ_TƯƠNG	CÔNG TY TNHH NARGI	2023-01-01	2400.0
KHOAI	CÔNG TY TNHH NARGI	2023-01-01	2200.0
MEN_VI_SINH	CÔNG TY VNIBIO	2023-01-25	420.0
NGÔ	CÔNG TY TNHH NARGI	2023-01-01	8400.0
SÁN	CÔNG TY TNHH NARGI	2023-01-01	2200.0
THÓC	CÔNG TY TNHH NARGI	2023-01-01	1000.0
BAO_BÌ_ĐÓNG_GÓI	CÔNG TY TNHH WOPACK	2023-02-22	2080.0
BỘT_ĐỒ_TƯƠNG	CÔNG TY TNHH NARGI	2023-02-22	3900.0
MEN_VI_SINH	CÔNG TY VNIBIO	2023-02-22	520.0

NHẬN XÉT MÔ HÌNH

Ưu điểm

- Giải quyết được yêu cầu bài toán với các giả thiết mà nhóm đặt ra
- Khá ngắn gọn và dễ hiểu
- Có tính tự động hóa
- Chọn lọc được những dữ liệu cần thiết, tránh rườm rà

Nhược điểm

- Chưa hoàn toàn giải quyết được nguyên tắc FEFO dẫn đến lãng phí nguyên liệu.
- Còn code thủ công, chưa tạo được vòng lặp hợp lý để tiết kiệm thời gian
- Chỉ phù hợp với những loại thành phẩm có LEADTIME sản xuất bằng 0 và LEADTIME đặt hàng bằng nhau.

THANK YOU!

