补货代码操作流程：

以要在2023年8月份安排补货为例

**首先先要构造出需求预测：**

1：首先对原始out\_bound文件进行处理，利用processing.py文件的make\_yanzheng函数：

1. **if** \_\_name\_\_=='\_\_main\_\_':
2. filename1 = r'F:\最新数据\v\_islm\_outbound\_20240220.csv'  # outbound文件
3. df = pd.read\_csv(filename1)
4. df['eta']=df['ship\_dt']
5. df=df.drop('ship\_dt', axis = 1)
7. mode='Day'#'Day' 'Week' 'Month'
8. sku=['customer\_name', 'customer\_part\_no','supplier\_name',  'supplier\_part\_no','manufacture\_name', 'site']
9. start='2022-11-01'
10. end='2023-8-14'
11. **print**(end)
12. df=make\_yanzheng(df,mode,sku,14,start,end)
13. df.to\_csv('f:\\最新数据\\out\_8\_14.csv')

这里end的截止日期为 2023-8-14。

2：得到处理好后的out\_bound文件，在tft\_multi\_dimension代码里进行模型训练，

1. **if** \_\_name\_\_=='\_\_main\_\_':
2. filename1=r'f:\\最新数据\out\_8\_14.csv'
3. con\_length=14
4. pre\_length=14
5. **print**(1)
6. sku=['customer\_name', 'customer\_part\_no','supplier\_name',   'supplier\_part\_no','manufacture\_name',  'site']
7. mode='Day'#'Week',’Month'
8. data=pd.read\_csv(filename1,index\_col=0)
9. filename2='E:\\model\\'
11. TFT(data,mode,sku,con\_length,pre\_length,filename2)

注意：该函数的倒数第三行

1. name=filename2+'tft'+'\_'+mode+'\_'+str(con\_length)+'\_'+str(pre\_length)+'buhuo.pkl'

其中的’buhuo.pkl’ 为参数文件的名字，可以适当标记修改。

3：进行预测，在pre\_multi\_dimension里进行预测：

1. **if** \_\_name\_\_=='\_\_main\_\_':
2. mode='Day'
3. con\_len=14
4. pre\_len=14
5. model='tft'
6. yanzheng=True
7. save\_model\_path='E:\\model\\'
8. sku=['customer\_name', 'customer\_part\_no','supplier\_name',   'supplier\_part\_no','manufacture\_name',  'site']
9. data=pd.read\_csv('f:\\最新数据\\out\_8\_14.csv')
10. # start = '2022-11-01'
11. end = '2023-8-14'
12. # data = make\_yanzheng(data, mode, sku, 14, 14, start, end)
13. **global** t
14. t=0
15. f=pre\_data=predict(data,model,mode,con\_len,pre\_len,save\_model\_path,sku,yanzheng,end)

注意：在函数里的第一二行：

1. name = save\_model\_path + model + '\_' + mode + '\_' + str(con\_len) + '\_' + str(pre\_len) + 'buhuo\_2.pkl'
2. name2=save\_model\_path + model + '\_' + mode + '\_' + str(con\_len) + '\_' + str(pre\_len) + 'buhuo\_pre2'+'.csv'

其中’buhuo2.pkl’是寻找名为’ ’buhuo2.pkl’参数文件，可根据训练时保存的

参数文件名字进行对应。

Name2里的'buhuo\_pre2'代表预测结果csv的名字，也可修改标记对应。

预测出8月1-14天的需求后。按照同样的方法预测出8月15-28天的需求。

同样的：

1：先对原始out\_bound文件进行处理，利用processing.py文件的make\_yanzheng函数：

这里修改

1. end='2023-8-14'
2. df.to\_csv('f:\\最新数据\\out\_8\_14.csv')

为

1. end='2023-8-28'
2. df.to\_csv('f:\\最新数据\\out\_8\_28.csv')

2：

得到处理好后的out\_bound文件，在tft\_multi\_dimension代码里进行模型训练，

这里需要修改读取的

1. filename1=r'f:\\最新数据\out\_8\_14.csv'

修改为：

1. filename1=r'f:\\最新数据\out\_8\_28.csv'

同时可以修改模型保存参数文件的名字。

3：进行预测，在pre\_multi\_dimension里进行预测：

同时需要修改读取的文件和end日期。以及函数前两行的读取参数文件的名字和预测文件输出结果的名字。

第一步完成后，得到两个预测文件

pre\_df1=pd.read\_csv(r'E:\model\tft\_Day\_14\_14buhuo\_pre1.csv')  
pre\_df2=pd.read\_csv(r'E:\model\tft\_Day\_14\_14buhuo\_pre2.csv')

以及 一个整理后的出库文件：

outbound\_df=pd.read\_csv(r'F:\最新数据\out\_8\_28.csv')

**补货预处理**

在buhuo\_process.py 文件里

1. pre\_df1=pd.read\_csv(r'E:\model\tft\_Day\_14\_14buhuo\_pre1.csv')
2. pre\_df2=pd.read\_csv(r'E:\model\tft\_Day\_14\_14buhuo\_pre2.csv')
3. inventory\_df=pd.read\_csv(r'F:\最新数据\v\_islm\_inventory\_20240220.csv')
4. inbound\_df=pd.read\_csv(r'F:\最新数据\inbound.csv')
5. asn\_df=pd.read\_csv(r'F:\最新数据\asn.csv')
6. outbound\_df=pd.read\_csv(r'F:\最新数据\out\_8\_28.csv')
7. sku=['supplier\_name', 'supplier\_part\_no', 'customer\_name', 'customer\_part\_no', 'manufacture\_name','site']
8. month=8
9. days=31
10. last\_day='2023-07-31'
11. save\_name='F://最新数据//'
12. first\_process(pre\_df1, pre\_df2, inventory\_df, inbound\_df, asn\_df, outbound\_df, sku, month, days, last\_day,
13. save\_name)

这里

1. month=8
2. days=31
3. last\_day='2023-07-31'

指的是 要安排补货的月份为8月，八月一共有31天，八月份的前一天为'2023-07-31' ,这一步是要筛选掉一些日期有误的数据，防止后续报错。

这一步骤执行完后，会保存

1. result\_df3.to\_csv(save\_name+'common\_outbound.csv')
2. result\_df2.to\_csv(save\_name+'common\_asn.csv')
3. result\_df1.to\_csv(save\_name+'common\_inbound.csv')
4. result\_df4.to\_csv(save\_name+'common\_inventory.csv')
5. result\_df5.to\_csv(save\_name+'common\_demand.csv')

分别是 出库，asn, 入库， 库存历史水平， 预测需求文件。

**执行补货：**

在buhuoyanzheng.py文件里：

1. **if** \_\_name\_\_=='\_\_main\_\_':
2. fahuo=20
3. baozhuang=20
4. df\_asn=pd.read\_csv(r'F:\补货实验\asn\_0.csv')
5. df\_inbound=pd.read\_csv(r'F:\补货实验\inbound\_0.csv')
6. df\_demand=pd.read\_csv(r'F:\补货实验\demand\_0.csv')
7. df\_inventory=pd.read\_csv(r'F:\补货实验\inventory\_0.csv')
8. df\_outbound=pd.read\_csv(r'F:\补货实验\out\_0.csv')
9. sku=['customer\_name',   'customer\_part\_no','supplier\_name', 'supplier\_part\_no', 'manufacture\_name']
10. rate = 1.4
11. df\_asn = df\_asn.sort\_values(sku)
12. df\_inbound = df\_inbound.sort\_values(sku)
13. df\_inventory = df\_inventory.sort\_values(sku)
14. df\_outbound = df\_outbound.sort\_values(sku)
15. df\_demand = df\_demand.sort\_values(sku)
17. df\_asn\_grouped = df\_asn.groupby(sku)
18. df\_inbound\_grouped = df\_inbound.groupby(sku)
19. df\_demand\_grouped = df\_demand.groupby(sku)
20. df\_inventory\_grouped = df\_inventory.groupby(sku)
21. df\_outbound\_grouped = df\_outbound.groupby(sku)
22. lower = []
23. df=pd.DataFrame()
24. i=0
25. **for** (group\_key1, group1), (group\_key2, group2) ,(group\_key3, group3),(group\_key4, group4),(group\_key5, group5)**in** zip(df\_asn\_grouped,df\_inbound\_grouped,df\_demand\_grouped,df\_inventory\_grouped,df\_outbound\_grouped):
26. buhuo\_df=buhuo(group1,group2,group3,group4,fahuo, baozhuang,group5,rate)
27. i+=1
28. **if** isinstance(buhuo\_df,bool):
29. buhuo\_df=pd.DataFrame()
30. df=pd.concat([df,buhuo\_df])
31. df.to\_csv(r'F:\补货实验\buhuo5.csv')

需要输入的参数有

1. df\_asn=pd.read\_csv(r'F:\补货实验\asn\_0.csv')
2. df\_inbound=pd.read\_csv(r'F:\补货实验\inbound\_0.csv')
3. df\_demand=pd.read\_csv(r'F:\补货实验\demand\_0.csv')
4. df\_inventory=pd.read\_csv(r'F:\补货实验\inventory\_0.csv')
5. df\_outbound=pd.read\_csv(r'F:\补货实验\out\_0.csv')

这是补货预处理执行后得到的五个文件，

然后最后一行

1. df.to\_csv(r'F:\补货实验\buhuo5.csv')

代表补货结果保存位置。