ip-project-supply-chain-management

November 11, 2023

1 SUPPLY CHAIN MANAGEMENT

INTRODUCTION

Supply chain management (SCM) is the process of managing the flow of goods and services to and from a business, including every step involved in turning raw materials and components into final products and getting them to the ultimate customer. Effective SCM can help streamline a company's activities to eliminate waste, maximize customer value, and gain a competitive advantage in the marketplace. Supply chain management (SCM) is the centralized management of the flow of goods and services to and from a company and includes all of the processes involved in transforming raw materials and components into final products.By managing the supply chain, companies can cut excess costs and deliver products to the consumer faster and more efficiently.Good supply chain management can help prevent expensive product recalls and lawsuits as well as bad publicity. The five most critical phases of SCM are planning, sourcing, production, distribution, and returns.A supply chain manager is tasked with controlling and reducing costs and avoiding supply shortages.

[4]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

2 Train Dataset

[5]:
```
df=pd.read_csv('supply_train.csv')
df
```

[5]:

	Unnamed: 0	Ware_house_ID	WH_Manager_ID	Location_type	WH_capacity_size	\
0	0	WH_100000	EID_50000	Urban	Small	
1	1	WH_100001	EID_50001	Rural	Large	
2	2	WH_100002	EID_50002	Rural	Mid	
3	3	WH_100003	EID_50003	Rural	Mid	
4	4	WH_100004	EID_50004	Rural	Large	
…	…	…	…	…	…	
16615	16615	WH_116615	EID_66615	Urban	Large	
16616	16616	WH_116616	EID_66616	Urban	Large	
16617	16617	WH_116617	EID_66617	Rural	Large	

```
16618       16618     WH_116618     EID_66618         Rural            Small
16619       16619     WH_116619     EID_66619         Rural            Large

        zone WH_regional_zone  num_refill_req_l3m  transport_issue_l1y  \
0       West           Zone 6                   3                    1
1      North           Zone 5                   0                    0
2      South           Zone 2                   1                    0
3      North           Zone 3                   7                    4
4      North           Zone 5                   3                    1
…      …               …                  …                  …
16615   West           Zone 6                   3                    1
16616  North           Zone 5                   2                    0
16617  North           Zone 6                   5                    0
16618   West           Zone 6                   3                    2
16619   West           Zone 5                   4                    0

       Competitor_in_mkt  …  electric_supply dist_from_hub  workers_num  \
0                      2  …                 1            91         29.0
1                      4  …                 1           210         31.0
2                      4  …                 0           161         37.0
3                      2  …                 0           103         21.0
4                      2  …                 1           112         25.0
…                    …  …               …           …          …
16615                  6  …                 0           240         19.0
16616                  2  …                 1           164         30.0
16617                  4  …                 1           211         24.0
16618                  3  …                 1           119          NaN
16619                  6  …                 1           261         34.0

       wh_est_year  storage_issue_reported_l3m  temp_reg_mach  \
0              NaN                          13              0
1              NaN                           4              0
2              NaN                          17              0
3              NaN                          17              1
4           2009.0                          18              0
…            …                         …           …
16615       2009.0                          14              0
16616          NaN                          17              0
16617       2003.0                          24              1
16618       2007.0                          16              0
16619       2001.0                          32              0

       approved_wh_govt_certificate  wh_breakdown_l3m  govt_check_l3m  \
0                                 A                 5              15
1                                 A                 3              17
2                                 A                 6              22
3                                A+                 3              27
```

```
4                                  C                 6              24
…                                  …                 …              …
16615                             B+                 5              23
16616                             B+                 6              24
16617                              B                 5              29
16618                              A                 5              15
16619                             B+                 4              10

       product_wg_ton
0               17115
1                5074
2               23137
3               22115
4               24071
…                   …
16615           16094
16616           21113
16617           28117
16618           21103
16619           38097

[16620 rows x 25 columns]
```

3 DATA EXPLORATION

[6]:
```
df.shape
```

[6]:
```
(16620, 25)
```

[7]:
```
df.columns
```

[7]:
```
Index(['Unnamed: 0', 'Ware_house_ID', 'WH_Manager_ID', 'Location_type',
       'WH_capacity_size', 'zone', 'WH_regional_zone', 'num_refill_req_l3m',
       'transport_issue_l1y', 'Competitor_in_mkt', 'retail_shop_num',
       'wh_owner_type', 'distributor_num', 'flood_impacted', 'flood_proof',
       'electric_supply', 'dist_from_hub', 'workers_num', 'wh_est_year',
       'storage_issue_reported_l3m', 'temp_reg_mach',
       'approved_wh_govt_certificate', 'wh_breakdown_l3m', 'govt_check_l3m',
       'product_wg_ton'],
      dtype='object')
```

[8]:
```
#Dropping unwanted column "Unnamed"
df=df.drop(columns=["Unnamed: 0"])
df
```

[8]:

	Ware_house_ID	WH_Manager_ID	Location_type	WH_capacity_size	zone	…
0	WH_100000	EID_50000	Urban	Small	West	
1	WH_100001	EID_50001	Rural	Large	North	
2	WH_100002	EID_50002	Rural	Mid	South	
3	WH_100003	EID_50003	Rural	Mid	North	
4	WH_100004	EID_50004	Rural	Large	North	
…	…	…	…	…	…	
16615	WH_116615	EID_66615	Urban	Large	West	
16616	WH_116616	EID_66616	Urban	Large	North	
16617	WH_116617	EID_66617	Rural	Large	North	
16618	WH_116618	EID_66618	Rural	Small	West	
16619	WH_116619	EID_66619	Rural	Large	West	

	WH_regional_zone	num_refill_req_l3m	transport_issue_l1y	…
0	Zone 6	3	1	
1	Zone 5	0	0	
2	Zone 2	1	0	
3	Zone 3	7	4	
4	Zone 5	3	1	
…	…	…	…	
16615	Zone 6	3	1	
16616	Zone 5	2	0	
16617	Zone 6	5	0	
16618	Zone 6	3	2	
16619	Zone 5	4	0	

	Competitor_in_mkt	retail_shop_num	…	electric_supply	dist_from_hub	…
0	2	4651	…	1	91	
1	4	6217	…	1	210	
2	4	4306	…	0	161	
3	2	6000	…	0	103	
4	2	4740	…	1	112	
…	…	…	…	…	…	
16615	6	4779	…	0	240	
16616	2	5718	…	1	164	
16617	4	4514	…	1	211	
16618	3	5829	…	1	119	
16619	6	3751	…	1	261	

	workers_num	wh_est_year	storage_issue_reported_l3m	temp_reg_mach	…
0	29.0	NaN	13	0	
1	31.0	NaN	4	0	
2	37.0	NaN	17	0	
3	21.0	NaN	17	1	
4	25.0	2009.0	18	0	
…	…	…	…	…	
16615	19.0	2009.0	14	0	

```
16616         30.0          NaN                          17              0
16617         24.0       2003.0                          24              1
16618          NaN       2007.0                          16              0
16619         34.0       2001.0                          32              0

       approved_wh_govt_certificate  wh_breakdown_l3m  govt_check_l3m  \
0                                 A                 5              15
1                                 A                 3              17
2                                 A                 6              22
3                                A+                 3              27
4                                 C                 6              24
…                               …                 …             …
16615                            B+                 5              23
16616                            B+                 6              24
16617                             B                 5              29
16618                             A                 5              15
16619                            B+                 4              10

       product_wg_ton
0               17115
1                5074
2               23137
3               22115
4               24071
…               …
16615           16094
16616           21113
16617           28117
16618           21103
16619           38097

[16620 rows x 24 columns]
```

[11]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16620 entries, 0 to 16619
Data columns (total 24 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Ware_house_ID                 16620 non-null  object
 1   WH_Manager_ID                 16620 non-null  object
 2   Location_type                 16620 non-null  object
 3   WH_capacity_size              16620 non-null  object
 4   zone                          16620 non-null  object
 5   WH_regional_zone              16620 non-null  object
 6   num_refill_req_l3m            16620 non-null  int64
```

```
 7   transport_issue_l1y           16620 non-null  int64
 8   Competitor_in_mkt             16620 non-null  int64
 9   retail_shop_num               16620 non-null  int64
 10  wh_owner_type                 16620 non-null  object
 11  distributor_num               16620 non-null  int64
 12  flood_impacted                16620 non-null  int64
 13  flood_proof                   16620 non-null  int64
 14  electric_supply               16620 non-null  int64
 15  dist_from_hub                 16620 non-null  int64
 16  workers_num                   15953 non-null  float64
 17  wh_est_year                   8760 non-null   float64
 18  storage_issue_reported_l3m    16620 non-null  int64
 19  temp_reg_mach                 16620 non-null  int64
 20  approved_wh_govt_certificate  16021 non-null  object
 21  wh_breakdown_l3m              16620 non-null  int64
 22  govt_check_l3m                16620 non-null  int64
 23  product_wg_ton                16620 non-null  int64
dtypes: float64(2), int64(14), object(8)
memory usage: 3.0+ MB
```

[9]:
```
df.describe()
```

[9]:

	num_refill_req_l3m	transport_issue_l1y	Competitor_in_mkt
count	16620.000000	16620.000000	16620.000000
mean	4.126655	0.780927	3.103129
std	2.606241	1.206351	1.147711
min	0.000000	0.000000	0.000000
25%	2.000000	0.000000	2.000000
50%	4.000000	0.000000	3.000000
75%	6.000000	1.000000	4.000000
max	8.000000	5.000000	12.000000

	retail_shop_num	distributor_num	flood_impacted	flood_proof
count	16620.000000	16620.000000	16620.000000	16620.000000
mean	4983.904994	42.473706	0.096871	0.056017
std	1051.032239	16.090000	0.295791	0.229961
min	1821.000000	15.000000	0.000000	0.000000
25%	4309.000000	29.000000	0.000000	0.000000
50%	4856.000000	42.000000	0.000000	0.000000
75%	5500.000000	56.000000	0.000000	0.000000
max	11008.000000	70.000000	1.000000	1.000000

	electric_supply	dist_from_hub	workers_num	wh_est_year
count	16620.000000	16620.000000	15953.000000	8760.000000
mean	0.655716	163.521901	28.911490	2009.345320
std	0.475149	62.701193	7.842046	7.544672
min	0.000000	55.000000	10.000000	1996.000000

```
25%           0.000000     109.000000     24.000000  2003.000000
50%           1.000000     164.000000     28.000000  2009.000000
75%           1.000000     218.000000     33.000000  2016.000000
max           1.000000     271.000000     98.000000  2023.000000

       storage_issue_reported_l3m  temp_reg_mach  wh_breakdown_l3m  \
count                16620.000000   16620.000000      16620.000000
mean                    17.127196       0.306137          3.491095
std                      9.163901       0.460901          1.688614
min                      0.000000       0.000000          0.000000
25%                     10.000000       0.000000          2.000000
50%                     18.000000       0.000000          3.000000
75%                     24.000000       1.000000          5.000000
max                     39.000000       1.000000          6.000000

       govt_check_l3m  product_wg_ton
count    16620.000000    16620.000000
mean        18.727377    22098.420096
std          8.619857    11620.337346
min          1.000000     2093.000000
25%         11.000000    13057.000000
50%         19.000000    22099.000000
75%         26.000000    30103.000000
max         32.000000    55151.000000
```

[7]:
```
df.dtypes
```

[7]:
```
SI.NO                          int64
Ware_house_ID                 object
WH_Manager_ID                 object
Location_type                 object
WH_capacity_size              object
zone                          object
WH_regional_zone              object
num_refill_req_l3m             int64
transport_issue_l1y            int64
Competitor_in_mkt              int64
retail_shop_num                int64
wh_owner_type                 object
distributor_num                int64
flood_impacted                 int64
flood_proof                    int64
electric_supply                int64
dist_from_hub                  int64
workers_num                  float64
wh_est_year                  float64
storage_issue_reported_l3m     int64
```

```
temp_reg_mach                       int64
approved_wh_govt_certificate       object
wh_breakdown_l3m                    int64
govt_check_l3m                      int64
product_wg_ton                      int64
dtype: object
```

4 Exploratory Data Analysis (EDA)

[12]:
```
#Distribution of categorical variables
categorical_columns=df.select_dtypes(include=['object'])
for col in categorical_columns:
    print(df[col].value_counts())
```

```
Ware_house_ID
WH_100000    1
WH_111101    1
WH_111071    1
WH_111072    1
WH_111073    1
            ..
WH_105545    1
WH_105546    1
WH_105547    1
WH_105548    1
WH_116619    1
Name: count, Length: 16620, dtype: int64
WH_Manager_ID
EID_50000    1
EID_61101    1
EID_61071    1
EID_61072    1
EID_61073    1
            ..
EID_55545    1
EID_55546    1
EID_55547    1
EID_55548    1
EID_66619    1
Name: count, Length: 16620, dtype: int64
Location_type
Rural    15272
Urban     1348
Name: count, dtype: int64
WH_capacity_size
Large    6743
```

```
Mid      6691
Small    3186
Name: count, dtype: int64
zone
North    6800
West     5320
South    4223
East      277
Name: count, dtype: int64
WH_regional_zone
Zone 6    5568
Zone 5    3027
Zone 4    2756
Zone 2    1998
Zone 3    1937
Zone 1    1334
Name: count, dtype: int64
wh_owner_type
Company Owned    8975
Rented           7645
Name: count, dtype: int64
approved_wh_govt_certificate
C     3638
B+    3260
B     3220
A     3132
A+    2771
Name: count, dtype: int64
```

[13]:
```
#Distribution of numerical variables
numerical_columns=df.select_dtypes(include=['int','float'])
for col in numerical_columns:
    print(df[col].value_counts())
```

```
num_refill_req_l3m
8    2032
3    1984
5    1967
7    1941
0    1900
4    1885
6    1857
1    1835
2    1219
Name: count, dtype: int64
transport_issue_l1y
0    10094
1     3072
```

```
2     1437
3     1272
4      508
5      237
Name: count, dtype: int64
Competitor_in_mkt
2     5790
3     4702
4     4438
5      825
6      375
1      291
7      136
8       49
10       6
9        6
12       1
0        1
Name: count, dtype: int64
retail_shop_num
4808    18
5022    17
4367    17
4611    16
4439    16
        ..
7176     1
3483     1
8522     1
7570     1
7347     1
Name: count, Length: 4356, dtype: int64
distributor_num
21    328
69    323
49    322
35    321
59    319
63    319
47    318
36    317
37    315
41    314
40    314
28    313
44    311
31    311
24    309
```

```
42    307
38    307
57    305
30    304
54    303
50    303
65    302
29    302
48    301
23    300
64    299
56    299
22    298
20    297
15    296
62    295
18    293
55    292
26    292
33    291
67    290
66    290
52    289
16    288
32    288
46    287
27    286
70    284
43    284
45    280
58    279
39    279
34    278
17    274
51    274
68    273
61    272
19    272
25    271
60    271
53    271
Name: count, dtype: int64
flood_impacted
0    15010
1     1610
Name: count, dtype: int64
flood_proof
0    15689
```

```
1      931
Name: count, dtype: int64
electric_supply
1    10898
0     5722
Name: count, dtype: int64
dist_from_hub
186    103
145     99
101     98
204     97
173     97
       …
130     60
85      60
95      58
138     56
194     53
Name: count, Length: 217, dtype: int64
workers_num
28.0    966
27.0    946
29.0    929
26.0    880
25.0    862
24.0    833
30.0    805
31.0    747
23.0    739
32.0    713
22.0    626
33.0    623
21.0    540
34.0    535
20.0    526
36.0    449
35.0    441
19.0    396
18.0    371
37.0    325
17.0    293
40.0    266
39.0    263
38.0    261
16.0    215
42.0    212
41.0    160
43.0    134
```

```
44.0    126
45.0    116
15.0     98
14.0     68
46.0     63
48.0     49
50.0     48
49.0     41
53.0     34
56.0     33
55.0     33
47.0     32
54.0     23
52.0     21
58.0     14
13.0     14
51.0     13
12.0     13
57.0     10
60.0      6
62.0      5
64.0      5
61.0      4
98.0      4
11.0      4
67.0      4
10.0      3
78.0      3
92.0      3
65.0      3
72.0      3
63.0      1
Name: count, dtype: int64
wh_est_year
2000.0    387
2014.0    365
2002.0    363
2004.0    357
2007.0    352
1998.0    346
2006.0    343
2016.0    342
2015.0    340
2020.0    337
2021.0    336
2010.0    335
2008.0    334
2012.0    331
```

```
2001.0    330
2005.0    330
2017.0    329
2019.0    328
2018.0    327
1999.0    323
2013.0    323
2009.0    319
2003.0    314
2011.0    314
1997.0    226
2022.0    212
1996.0    126
2023.0     91
Name: count, dtype: int64
storage_issue_reported_l3m
24    956
5     935
25    855
18    728
4     717
20    698
6     688
19    685
16    627
22    606
0     599
11    595
15    595
23    591
14    547
17    510
12    503
9     500
13    470
21    462
10    408
26    387
27    380
7     333
8     260
30    231
28    221
29    209
31    200
32    195
33    190
34    180
```

```
38    130
35    125
36    107
39    102
37     95
Name: count, dtype: int64
temp_reg_mach
0    11532
1     5088
Name: count, dtype: int64
wh_breakdown_l3m
2    3354
3    3322
4    2688
6    2667
5    2651
1    1339
0     599
Name: count, dtype: int64
govt_check_l3m
26    1905
23    1220
19    1129
28     966
14     952
6      819
27     808
11     775
9      660
32     633
12     632
29     605
10     598
25     557
15     465
21     429
24     426
1      377
17     332
3      298
13     281
2      279
30     251
31     224
22     200
8      167
5      162
18     156
```

```
16     132
20      76
4       63
7       43
Name: count, dtype: int64
product_wg_ton
6081     19
6057     15
6107     14
5139     13
6097     13
         ..
49072     1
41122     1
40128     1
54144     1
21069     1
Name: count, Length: 4248, dtype: int64
```

5 Data Preprocessing

[14]:
```
#finding missing values
df.isnull().sum()
```

[14]:
```
Ware_house_ID                    0
WH_Manager_ID                    0
Location_type                    0
WH_capacity_size                 0
zone                             0
WH_regional_zone                 0
num_refill_req_l3m               0
transport_issue_l1y              0
Competitor_in_mkt                0
retail_shop_num                  0
wh_owner_type                    0
distributor_num                  0
flood_impacted                   0
flood_proof                      0
electric_supply                  0
dist_from_hub                    0
workers_num                    667
wh_est_year                   7860
storage_issue_reported_l3m       0
temp_reg_mach                    0
approved_wh_govt_certificate   599
wh_breakdown_l3m                 0
govt_check_l3m                   0
```

```
product_wg_ton                    0
dtype: int64
```

[15]:
```
# Filling missing values with the median
df['workers_num']=df['workers_num'].fillna(df['workers_num'].median())
df['wh_est_year']=df['wh_est_year'].fillna(df['wh_est_year'].median())
# Filling missing values with a default category or mode
df['approved_wh_govt_certificate'].fillna('NIL', inplace=True)
df.isnull().sum()
```

[15]:
```
Ware_house_ID                  0
WH_Manager_ID                  0
Location_type                  0
WH_capacity_size               0
zone                           0
WH_regional_zone               0
num_refill_req_l3m             0
transport_issue_l1y            0
Competitor_in_mkt              0
retail_shop_num                0
wh_owner_type                  0
distributor_num                0
flood_impacted                 0
flood_proof                    0
electric_supply                0
dist_from_hub                  0
workers_num                    0
wh_est_year                    0
storage_issue_reported_l3m     0
temp_reg_mach                  0
approved_wh_govt_certificate   0
wh_breakdown_l3m               0
govt_check_l3m                 0
product_wg_ton                 0
dtype: int64
```

[16]:
```
# Remove duplicate rows
df.drop_duplicates(inplace=True)
df
```

[16]:

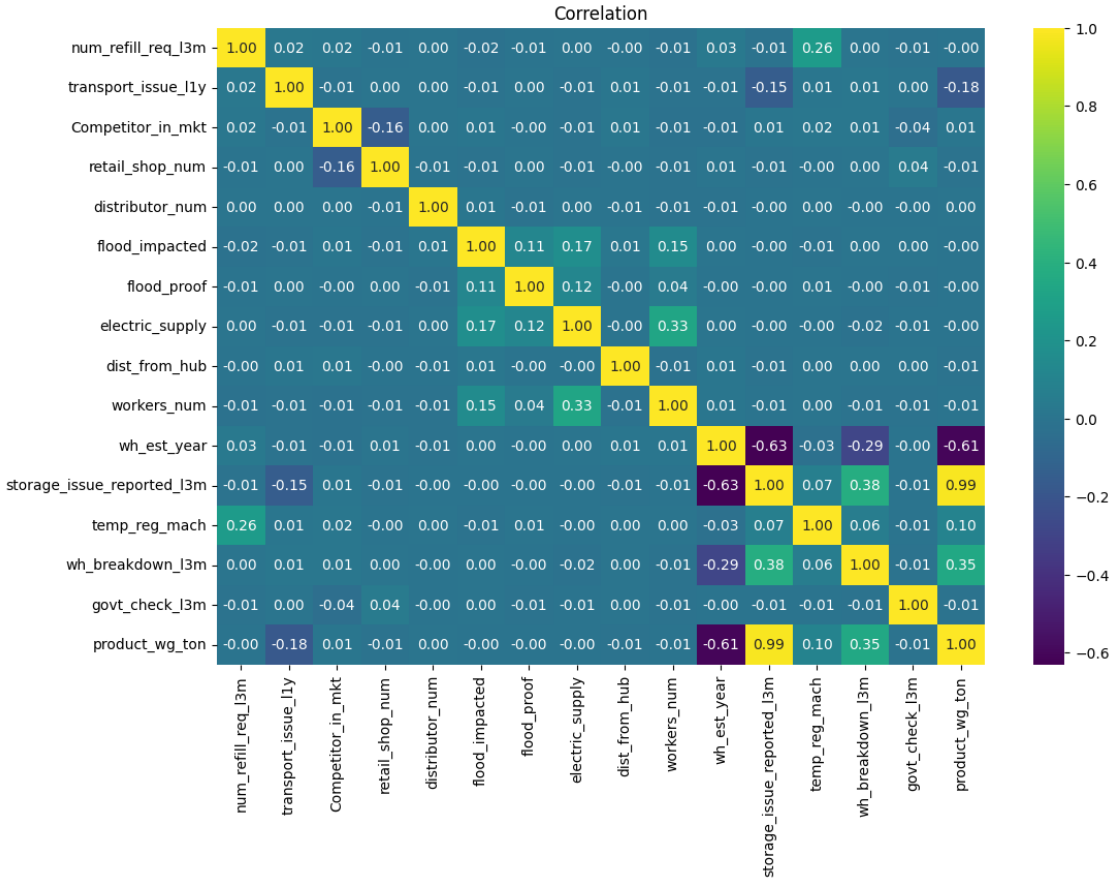
	Ware_house_ID	WH_Manager_ID	Location_type	WH_capacity_size	zone	\
0	WH_100000	EID_50000	Urban	Small	West	
1	WH_100001	EID_50001	Rural	Large	North	
2	WH_100002	EID_50002	Rural	Mid	South	
3	WH_100003	EID_50003	Rural	Mid	North	
4	WH_100004	EID_50004	Rural	Large	North	
…	…	…	…	…	…	

```
16615     WH_116615     EID_66615         Urban            Large   West
16616     WH_116616     EID_66616         Urban            Large  North
16617     WH_116617     EID_66617         Rural            Large  North
16618     WH_116618     EID_66618         Rural            Small   West
16619     WH_116619     EID_66619         Rural            Large   West

      WH_regional_zone  num_refill_req_l3m  transport_issue_l1y  \
0               Zone 6                   3                    1
1               Zone 5                   0                    0
2               Zone 2                   1                    0
3               Zone 3                   7                    4
4               Zone 5                   3                    1
…                …                   …                   …
16615           Zone 6                   3                    1
16616           Zone 5                   2                    0
16617           Zone 6                   5                    0
16618           Zone 6                   3                    2
16619           Zone 5                   4                    0

       Competitor_in_mkt  retail_shop_num  … electric_supply  dist_from_hub  \
0                      2             4651  …                1             91
1                      4             6217  …                1            210
2                      4             4306  …                0            161
3                      2             6000  …                0            103
4                      2             4740  …                1            112
…                  …             …  …            …            …
16615                  6             4779  …                0            240
16616                  2             5718  …                1            164
16617                  4             4514  …                1            211
16618                  3             5829  …                1            119
16619                  6             3751  …                1            261

       workers_num  wh_est_year  storage_issue_reported_l3m  temp_reg_mach  \
0             29.0       2009.0                          13              0
1             31.0       2009.0                           4              0
2             37.0       2009.0                          17              0
3             21.0       2009.0                          17              1
4             25.0       2009.0                          18              0
…            …          …                        …            …
16615         19.0       2009.0                          14              0
16616         30.0       2009.0                          17              0
16617         24.0       2003.0                          24              1
16618         28.0       2007.0                          16              0
16619         34.0       2001.0                          32              0

       approved_wh_govt_certificate  wh_breakdown_l3m  govt_check_l3m  \
0                                 A                 5              15
```

```
1                                  A                3              17
2                                  A                6              22
3                                 A+                3              27
4                                  C                6              24
…                                  …                …               …
16615                             B+                5              23
16616                             B+                6              24
16617                              B                5              29
16618                              A                5              15
16619                             B+                4              10

       product_wg_ton
0               17115
1                5074
2               23137
3               22115
4               24071
…                   …
16615           16094
16616           21113
16617           28117
16618           21103
16619           38097

[16620 rows x 24 columns]
```

6 Correlation Matrix

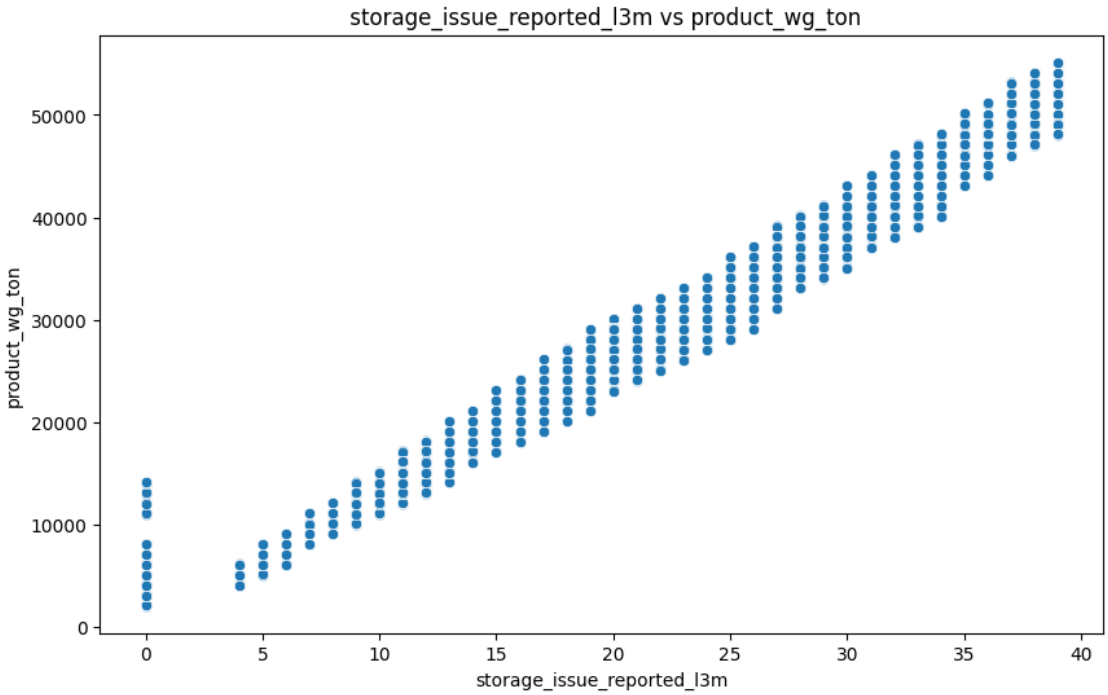
[17]:
```
# Select only numerical columns for correlation analysis
numerical_columns=df.select_dtypes(include=['int64', 'float64'])

# Calculate the correlation matrix for numerical columns
corr_matrix=numerical_columns.corr()

# Create a heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(corr_matrix, annot=True, cmap='viridis', fmt=".2f")
plt.title('Correlation')
plt.show()
```



Correlation heatmap shows the relationships between numerical variables. There is a strong positive correlation between storage_issue_reported_l3m and product_wg_ton.

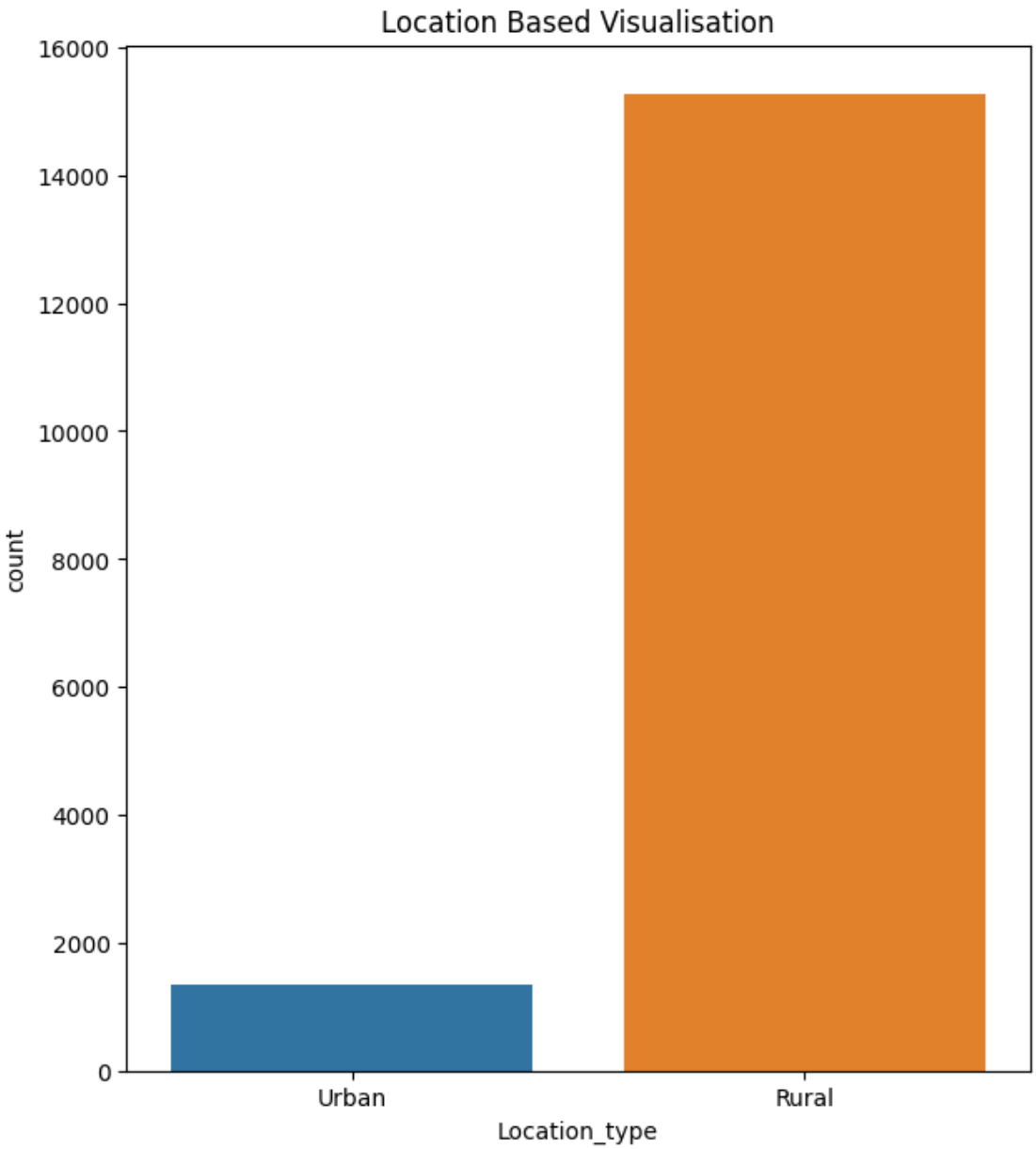
7 Data Visualization

[19]:
```
#Scatter plot with storage_issue_reported_l3m vs product_wg_ton which is␣
 ↪showing strong correlation
plt.figure(figsize=(10,6))
sns.scatterplot(x="storage_issue_reported_l3m",y="product_wg_ton",data=df)
plt.title(" storage_issue_reported_l3m vs product_wg_ton")
plt.xlabel("storage_issue_reported_l3m")
plt.ylabel("product_wg_ton")
plt.show()
```



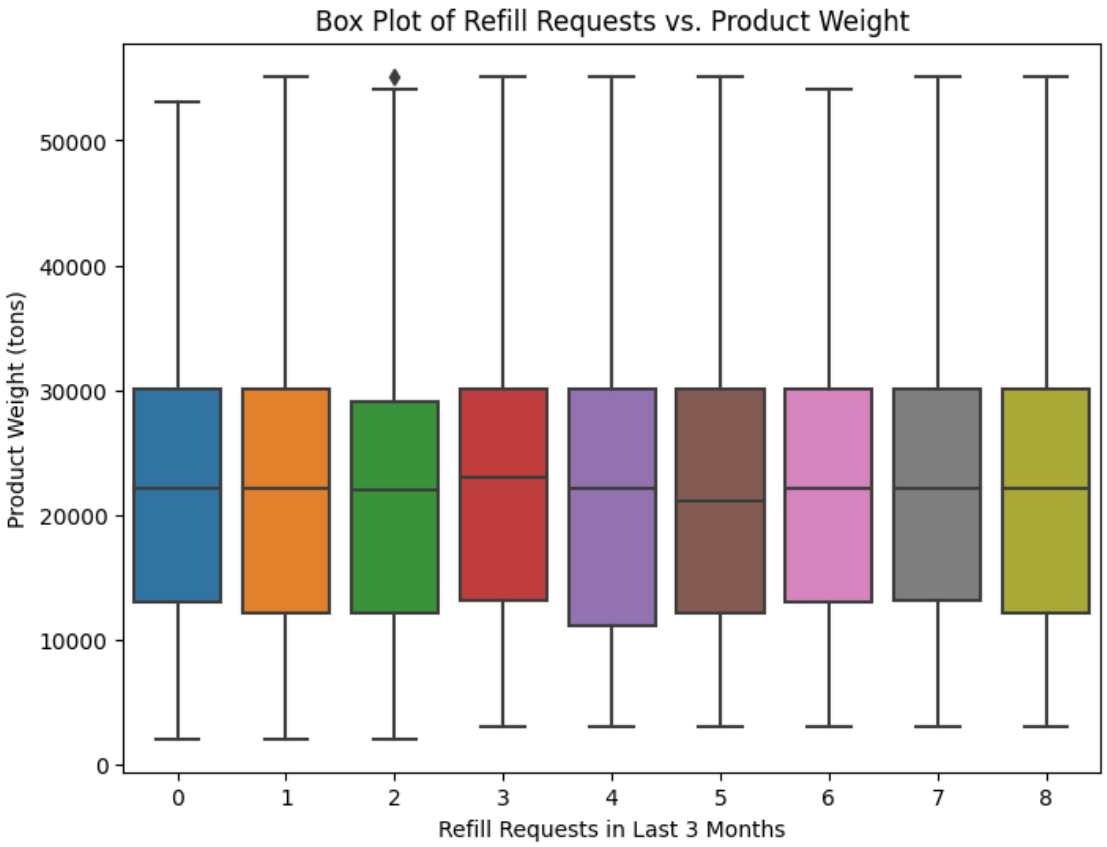
The data points are clustered close together and form a line that slopes upwards, indicating that there is a strong positive correlation between storage_issue_reported_l3m and product_wg_ton. This means that as storage_issue_reported_l3m increases, product_wg_ton also tends to increases.

[20]:
```
plt.figure(figsize=(7,8))
sns.countplot(x='Location_type',data=df)
plt.title("Location Based Visualisation")
plt.show()
```



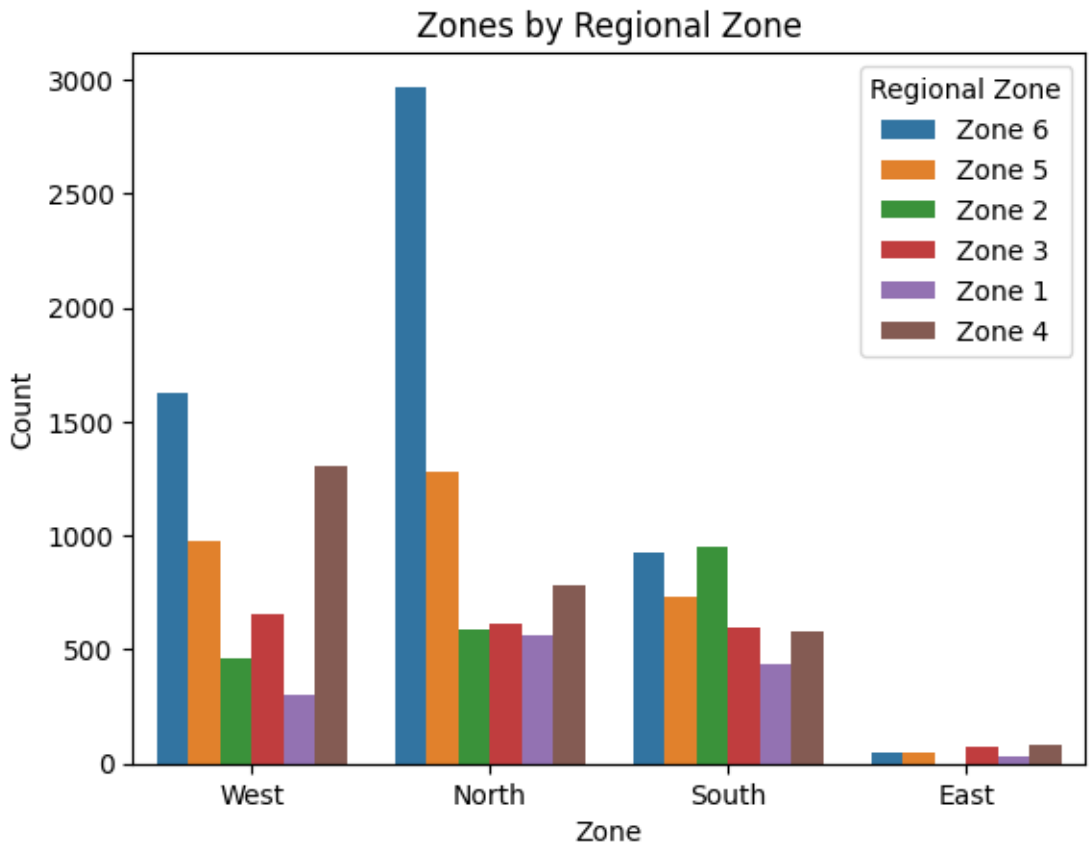
Urban locations have smaller warehouses compared to rural areas.

[21]:
```
# Explore relationships between numerical variables and the target variable
plt.figure(figsize=(8, 6))
sns.boxplot(x='num_refill_req_l3m', y='product_wg_ton', data=df)
plt.title('Box Plot of Refill Requests vs. Product Weight')
plt.xlabel('Refill Requests in Last 3 Months')
plt.ylabel('Product Weight (tons)')
plt.show()
```



From this boxplot we cannot find any strong relationships between "num_refill_req_l3m" and "product_wg_ton."

[22]:
```
# Explore the distribution of 'zone' and 'WH_regional_zone'
plt.figure
sns.countplot(data=df, x='zone', hue='WH_regional_zone')
plt.title('Zones by Regional Zone')
plt.xlabel('Zone')
plt.ylabel('Count')
plt.legend(title='Regional Zone')
plt.show()
```



This countplot shows the distributions of warehouse zone across different regional zones.Regional zone6 is showing the highest count of warehouses.

8 Data Encoding

[23]:
```
# Filter the columns with non-numeric data types
categorical_columns = df.select_dtypes(include=['object']).columns
categorical_columns
```

[23]:
```
Index(['Ware_house_ID', 'WH_Manager_ID', 'Location_type', 'WH_capacity_size',
       'zone', 'WH_regional_zone', 'wh_owner_type',
       'approved_wh_govt_certificate'],
      dtype='object')
```

[24]:
```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()

# Convert variables using Label Encoding
label_encoder = LabelEncoder()
```

```
categorical_variables = ['Location_type', 'WH_capacity_size',
       'zone', 'WH_regional_zone', 'wh_owner_type',
       'approved_wh_govt_certificate']
for col in categorical_variables:
    df[col] = label_encoder.fit_transform(df[col])
df
```

[24]:

```
      Ware_house_ID WH_Manager_ID  Location_type  WH_capacity_size  zone  \
0         WH_100000     EID_50000              1                 2     3
1         WH_100001     EID_50001              0                 0     1
2         WH_100002     EID_50002              0                 1     2
3         WH_100003     EID_50003              0                 1     1
4         WH_100004     EID_50004              0                 0     1
…               …             …              …                 …     …
16615     WH_116615     EID_66615              1                 0     3
16616     WH_116616     EID_66616              1                 0     1
16617     WH_116617     EID_66617              0                 0     1
16618     WH_116618     EID_66618              0                 2     3
16619     WH_116619     EID_66619              0                 0     3

       WH_regional_zone  num_refill_req_l3m  transport_issue_l1y  \
0                     5                   3                    1
1                     4                   0                    0
2                     1                   1                    0
3                     2                   7                    4
4                     4                   3                    1
…                   …                   …                    …
16615                 5                   3                    1
16616                 4                   2                    0
16617                 5                   5                    0
16618                 5                   3                    2
16619                 4                   4                    0

       Competitor_in_mkt  retail_shop_num  …  electric_supply  \
0                      2             4651  …                1
1                      4             6217  …                1
2                      4             4306  …                0
3                      2             6000  …                0
4                      2             4740  …                1
…                    …                …  …              …
16615                  6             4779  …                0
16616                  2             5718  …                1
16617                  4             4514  …                1
16618                  3             5829  …                1
16619                  6             3751  …                1

       dist_from_hub  workers_num  wh_est_year  storage_issue_reported_l3m  \
```

```
0                91         29.0       2009.0                          13
1               210         31.0       2009.0                           4
2               161         37.0       2009.0                          17
3               103         21.0       2009.0                          17
4               112         25.0       2009.0                          18
…               …           …          …                    …
16615           240         19.0       2009.0                          14
16616           164         30.0       2009.0                          17
16617           211         24.0       2003.0                          24
16618           119         28.0       2007.0                          16
16619           261         34.0       2001.0                          32

       temp_reg_mach  approved_wh_govt_certificate  wh_breakdown_l3m  \
0                  0                             0                 5
1                  0                             0                 3
2                  0                             0                 6
3                  1                             1                 3
4                  0                             4                 6
…              …                         …              …
16615              0                             3                 5
16616              0                             3                 6
16617              1                             2                 5
16618              0                             0                 5
16619              0                             3                 4

       govt_check_l3m  product_wg_ton
0                  15           17115
1                  17            5074
2                  22           23137
3                  27           22115
4                  24           24071
…              …             …
16615              23           16094
16616              24           21113
16617              29           28117
16618              15           21103
16619              10           38097

[16620 rows x 24 columns]
```

[25]:
```
#Removing unwanted columns for creating machine learning models.
df1=df.drop(["Ware_house_ID","WH_Manager_ID"],axis=1)
df1
```

[25]:
```
       Location_type  WH_capacity_size  zone  WH_regional_zone  \
0                  1                 2     3                 5
1                  0                 0     1                 4
```

```
2                  0                  1     2                  1
3                  0                  1     1                  2
4                  0                  0     1                  4
…                  …                  …     …                  …
16615              1                  0     3                  5
16616              1                  0     1                  4
16617              0                  0     1                  5
16618              0                  2     3                  5
16619              0                  0     3                  4

       num_refill_req_l3m  transport_issue_l1y  Competitor_in_mkt  \
0                       3                     1                  2
1                       0                     0                  4
2                       1                     0                  4
3                       7                     4                  2
4                       3                     1                  2
…                       …                     …                  …
16615                   3                     1                  6
16616                   2                     0                  2
16617                   5                     0                  4
16618                   3                     2                  3
16619                   4                     0                  6

       retail_shop_num  wh_owner_type  distributor_num  …  electric_supply  \
0                 4651              1               24  …                1
1                 6217              0               47  …                1
2                 4306              0               64  …                0
3                 6000              1               50  …                0
4                 4740              0               42  …                1
…                 …                 …               …   …                …
16615             4779              1               70  …                0
16616             5718              0               26  …                1
16617             4514              0               50  …                1
16618             5829              1               59  …                1
16619             3751              1               49  …                1

       dist_from_hub  workers_num  wh_est_year  storage_issue_reported_l3m  \
0                 91         29.0       2009.0                          13
1                210         31.0       2009.0                           4
2                161         37.0       2009.0                          17
3                103         21.0       2009.0                          17
4                112         25.0       2009.0                          18
…                …           …          …                               …
16615            240         19.0       2009.0                          14
16616            164         30.0       2009.0                          17
16617            211         24.0       2003.0                          24
16618            119         28.0       2007.0                          16
```

```
16619            261         34.0       2001.0                          32

       temp_reg_mach  approved_wh_govt_certificate  wh_breakdown_l3m  \
0                  0                             0                 5
1                  0                             0                 3
2                  0                             0                 6
3                  1                             1                 3
4                  0                             4                 6
…                …                             …               …
16615              0                             3                 5
16616              0                             3                 6
16617              1                             2                 5
16618              0                             0                 5
16619              0                             3                 4

       govt_check_l3m  product_wg_ton
0                  15           17115
1                  17            5074
2                  22           23137
3                  27           22115
4                  24           24071
…                …              …
16615              23           16094
16616              24           21113
16617              29           28117
16618              15           21103
16619              10           38097

[16620 rows x 22 columns]
```

9 Train Test Split

[26]:
```
X_train=df1.drop(["product_wg_ton"],axis=1)
X_train.shape
```

[26]:
```
(16620, 21)
```

[27]:
```
y_train=df1['product_wg_ton']
y_train
```

[27]:
```
0        17115
1         5074
2        23137
3        22115
4        24071
         …
```

```
16615    16094
16616    21113
16617    28117
16618    21103
16619    38097
Name: product_wg_ton, Length: 16620, dtype: int64
```

10 Test Dataset

[28]:
```
#Data Preprocessing
df2=pd.read_csv("supply_test.csv")
df2
```

[28]:

	Unnamed: 0	Ware_house_ID	WH_Manager_ID	Location_type	WH_capacity_size	zone	WH_regional_zone	num_refill_req_l3m	transport_issue_l1y	Competitor_in_mkt	…	electric_supply	dist_from_hub	workers_num
0	16621	WH_116621	EID_66621	Rural	Large	North	Zone 5	5	0	3	…	0	156	30.0
1	16622	WH_116622	EID_66622	Rural	Large	North	Zone 5	5	0	2	…	1	79	31.0
2	16623	WH_116623	EID_66623	Rural	Small	North	Zone 6	3	0	3	…	1	70	41.0
3	16624	WH_116624	EID_66624	Rural	Mid	West	Zone 4	5	2	2	…	1	255	33.0
4	16625	WH_116625	EID_66625	Urban	Mid	North	Zone 4	6	0	4	…	0	205	20.0
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
5524	22145	WH_122145	EID_72145	Rural	Large	North	Zone 6	8	1					
5525	22146	WH_122146	EID_72146	Rural	Small	South	Zone 6	3	1					
5526	22147	WH_122147	EID_72147	Rural	Large	North	Zone 6	7	2					
5527	22148	WH_122148	EID_72148	Rural	Large	North	Zone 6	6	2					
5528	22149	WH_122149	EID_72149	Rural	Mid	East	Zone 4	7	2					

```
5524                3  …                1          203         28.0
5525                3  …                0          170          NaN
5526                2  …                0           99         22.0
5527                4  …                0          220         17.0
5528                9  …                1          104         36.0

      wh_est_year  storage_issue_reported_l3m  temp_reg_mach  \
0          2006.0                          24              0
1          2019.0                           5              1
2          2008.0                          19              1
3          2017.0                           9              1
4          1999.0                          25              0
…           …                          …           …
5524       2007.0                          22              1
5525          NaN                          26              0
5526          NaN                          20              0
5527          NaN                          13              0
5528          NaN                          11              0

      approved_wh_govt_certificate  wh_breakdown_l3m  govt_check_l3m  \
0                                A                 2               5
1                                C                 2              24
2                               A+                 5               9
3                               A+                 3              11
4                                B                 4              26
…                              …                 …           …
5524                            A+                 2               4
5525                             C                 5               9
5526                            B+                 6               4
5527                             B                 3              28
5528                             C                 6              26

      product_wg_ton
0              30132
1               6075
2              24076
3              13092
4              29071
…              …
5524           29138
5525           33108
5526           24072
5527           15055
5528           14103

[5529 rows x 25 columns]
```

[29]:
```
#Dropping unwanted column "Unnamed"
df2=df2.drop(columns=["Unnamed: 0"])
df2
```

[29]:

	Ware_house_ID	WH_Manager_ID	Location_type	WH_capacity_size	zone
0	WH_116621	EID_66621	Rural	Large	North
1	WH_116622	EID_66622	Rural	Large	North
2	WH_116623	EID_66623	Rural	Small	North
3	WH_116624	EID_66624	Rural	Mid	West
4	WH_116625	EID_66625	Urban	Mid	North
…	…	…	…	…	…
5524	WH_122145	EID_72145	Rural	Large	North
5525	WH_122146	EID_72146	Rural	Small	South
5526	WH_122147	EID_72147	Rural	Large	North
5527	WH_122148	EID_72148	Rural	Large	North
5528	WH_122149	EID_72149	Rural	Mid	East

	WH_regional_zone	num_refill_req_l3m	transport_issue_l1y
0	Zone 5	5	0
1	Zone 5	5	0
2	Zone 6	3	0
3	Zone 4	5	2
4	Zone 4	6	0
…	…	…	…
5524	Zone 6	8	1
5525	Zone 6	3	1
5526	Zone 6	7	2
5527	Zone 6	6	2
5528	Zone 4	7	2

	Competitor_in_mkt	retail_shop_num	…	electric_supply	dist_from_hub
0	3	5590	…	0	156
1	2	5856	…	1	79
2	3	4803	…	1	70
3	2	4784	…	1	255
4	4	3699	…	0	205
…	…	…	…	…	…
5524	3	5030	…	1	203
5525	3	4320	…	0	170
5526	2	5268	…	0	99
5527	4	4378	…	0	220
5528	9	3626	…	1	104

	workers_num	wh_est_year	storage_issue_reported_l3m	temp_reg_mach
0	30.0	2006.0	24	0
1	31.0	2019.0	5	1
2	41.0	2008.0	19	1

```
3            33.0       2017.0                          9              1
4            20.0       1999.0                         25              0
…             …          …                              …              …
5524         28.0       2007.0                         22              1
5525          NaN          NaN                         26              0
5526         22.0          NaN                         20              0
5527         17.0          NaN                         13              0
5528         36.0          NaN                         11              0

      approved_wh_govt_certificate  wh_breakdown_l3m  govt_check_l3m  \
0                                A                 2               5
1                                C                 2              24
2                               A+                 5               9
3                               A+                 3              11
4                                B                 4              26
…                              …                  …              …
5524                            A+                 2               4
5525                             C                 5               9
5526                            B+                 6               4
5527                             B                 3              28
5528                             C                 6              26

      product_wg_ton
0              30132
1               6075
2              24076
3              13092
4              29071
…              …
5524           29138
5525           33108
5526           24072
5527           15055
5528           14103

[5529 rows x 24 columns]
```

[30]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5529 entries, 0 to 5528
Data columns (total 24 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Ware_house_ID               5529 non-null   object
 1   WH_Manager_ID               5529 non-null   object
 2   Location_type               5529 non-null   object
```

```
 3   WH_capacity_size              5529 non-null   object
 4   zone                          5529 non-null   object
 5   WH_regional_zone              5529 non-null   object
 6   num_refill_req_l3m            5529 non-null   int64
 7   transport_issue_l1y           5529 non-null   int64
 8   Competitor_in_mkt             5529 non-null   int64
 9   retail_shop_num               5529 non-null   int64
 10  wh_owner_type                 5529 non-null   object
 11  distributor_num               5529 non-null   int64
 12  flood_impacted                5529 non-null   int64
 13  flood_proof                   5529 non-null   int64
 14  electric_supply               5529 non-null   int64
 15  dist_from_hub                 5529 non-null   int64
 16  workers_num                   5319 non-null   float64
 17  wh_est_year                   2844 non-null   float64
 18  storage_issue_reported_l3m    5529 non-null   int64
 19  temp_reg_mach                 5529 non-null   int64
 20  approved_wh_govt_certificate  5323 non-null   object
 21  wh_breakdown_l3m              5529 non-null   int64
 22  govt_check_l3m                5529 non-null   int64
 23  product_wg_ton                5529 non-null   int64
dtypes: float64(2), int64(14), object(8)
memory usage: 1.0+ MB
```

[31]:
```
df2.describe()
```

[31]:

	num_refill_req_l3m	transport_issue_l1y	Competitor_in_mkt
count	5529.000000	5529.000000	5529.000000
mean	4.007235	0.765600	3.106167
std	2.604325	1.187567	1.128396
min	0.000000	0.000000	1.000000
25%	2.000000	0.000000	2.000000
50%	4.000000	0.000000	3.000000
75%	6.000000	1.000000	4.000000
max	8.000000	5.000000	9.000000

	retail_shop_num	distributor_num	flood_impacted	flood_proof
count	5529.000000	5529.000000	5529.000000	5529.000000
mean	4980.695424	42.128052	0.104178	0.049919
std	1049.617325	15.959934	0.305519	0.217797
min	1953.000000	15.000000	0.000000	0.000000
25%	4310.000000	28.000000	0.000000	0.000000
50%	4863.000000	42.000000	0.000000	0.000000
75%	5492.000000	56.000000	0.000000	0.000000
max	10846.000000	70.000000	1.000000	1.000000

	electric_supply	dist_from_hub	workers_num	wh_est_year

```
count      5529.000000    5529.000000  5319.000000  2844.000000
mean          0.657262     163.899982    29.010528  2009.572785
std           0.474668      62.544704     7.848478     7.473201
min           0.000000      55.000000    10.000000  1996.000000
25%           0.000000     110.000000    24.000000  2003.000000
50%           1.000000     165.000000    28.000000  2010.000000
75%           1.000000     218.000000    33.000000  2016.000000
max           1.000000     271.000000    98.000000  2023.000000

       storage_issue_reported_l3m  temp_reg_mach  wh_breakdown_l3m  \
count                 5529.000000    5529.000000       5529.000000
mean                    17.085549       0.298426          3.477302
std                      9.206551       0.457609          1.700717
min                      0.000000       0.000000          0.000000
25%                     10.000000       0.000000          2.000000
50%                     17.000000       0.000000          3.000000
75%                     24.000000       1.000000          5.000000
max                     39.000000       1.000000          6.000000

       govt_check_l3m  product_wg_ton
count     5529.000000     5529.000000
mean        18.892205    22052.334599
std          8.716737    11645.738485
min          1.000000     2065.000000
25%         11.000000    12143.000000
50%         21.000000    22099.000000
75%         26.000000    30099.000000
max         32.000000    55144.000000
```

[32]:
```
df2.dtypes
```

[32]:
```
Ware_house_ID                object
WH_Manager_ID                object
Location_type                object
WH_capacity_size             object
zone                         object
WH_regional_zone             object
num_refill_req_l3m            int64
transport_issue_l1y           int64
Competitor_in_mkt             int64
retail_shop_num               int64
wh_owner_type                object
distributor_num               int64
flood_impacted                int64
flood_proof                   int64
electric_supply               int64
dist_from_hub                 int64
```

```
workers_num                      float64
wh_est_year                      float64
storage_issue_reported_l3m         int64
temp_reg_mach                      int64
approved_wh_govt_certificate      object
wh_breakdown_l3m                   int64
govt_check_l3m                     int64
product_wg_ton                     int64
dtype: object
```

[33]:
```
df2.isnull().sum()
```

[33]:
```
Ware_house_ID                    0
WH_Manager_ID                    0
Location_type                    0
WH_capacity_size                 0
zone                             0
WH_regional_zone                 0
num_refill_req_l3m               0
transport_issue_l1y              0
Competitor_in_mkt                0
retail_shop_num                  0
wh_owner_type                    0
distributor_num                  0
flood_impacted                   0
flood_proof                      0
electric_supply                  0
dist_from_hub                    0
workers_num                    210
wh_est_year                   2685
storage_issue_reported_l3m       0
temp_reg_mach                    0
approved_wh_govt_certificate   206
wh_breakdown_l3m                 0
govt_check_l3m                   0
product_wg_ton                   0
dtype: int64
```

[34]:
```
# Filling missing values with the median
df2['workers_num']=df2['workers_num'].fillna(df2['workers_num'].median())
df2['wh_est_year']=df2['wh_est_year'].fillna(df2['wh_est_year'].median())
# Filling missing values with a default category or mode
df2['approved_wh_govt_certificate'].fillna('NIL', inplace=True)
df2.isnull().sum()
```

[34]:
```
Ware_house_ID                  0
WH_Manager_ID                  0
```

```
Location_type                   0
WH_capacity_size                0
zone                            0
WH_regional_zone                0
num_refill_req_l3m              0
transport_issue_l1y             0
Competitor_in_mkt               0
retail_shop_num                 0
wh_owner_type                   0
distributor_num                 0
flood_impacted                  0
flood_proof                     0
electric_supply                 0
dist_from_hub                   0
workers_num                     0
wh_est_year                     0
storage_issue_reported_l3m      0
temp_reg_mach                   0
approved_wh_govt_certificate    0
wh_breakdown_l3m                0
govt_check_l3m                  0
product_wg_ton                  0
dtype: int64
```

[35]:
```
#Data Encoding
# Filter the columns with non-numeric data types
categorical_columns = df2.select_dtypes(include=['object']).columns
categorical_columns
```

[35]:
```
Index(['Ware_house_ID', 'WH_Manager_ID', 'Location_type', 'WH_capacity_size',
       'zone', 'WH_regional_zone', 'wh_owner_type',
       'approved_wh_govt_certificate'],
      dtype='object')
```

[36]:
```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()

# Convert variables using Label Encoding
label_encoder = LabelEncoder()
categorical_variables = ['Location_type', 'WH_capacity_size',
       'zone', 'WH_regional_zone', 'wh_owner_type',
       'approved_wh_govt_certificate']
for col in categorical_variables:
    df2[col] = label_encoder.fit_transform(df2[col])
df2
```

[36]:

	Ware_house_ID	WH_Manager_ID	Location_type	WH_capacity_size	zone	…
0	WH_116621	EID_66621	0	0	1	
1	WH_116622	EID_66622	0	0	1	
2	WH_116623	EID_66623	0	2	1	
3	WH_116624	EID_66624	0	1	3	
4	WH_116625	EID_66625	1	1	1	
…	…	…	…	…	…	
5524	WH_122145	EID_72145	0	0	1	
5525	WH_122146	EID_72146	0	2	2	
5526	WH_122147	EID_72147	0	0	1	
5527	WH_122148	EID_72148	0	0	1	
5528	WH_122149	EID_72149	0	1	0	

	WH_regional_zone	num_refill_req_l3m	transport_issue_l1y	…
0	4	5	0	
1	4	5	0	
2	5	3	0	
3	3	5	2	
4	3	6	0	
…	…	…	…	
5524	5	8	1	
5525	5	3	1	
5526	5	7	2	
5527	5	6	2	
5528	3	7	2	

	Competitor_in_mkt	retail_shop_num	…	electric_supply	dist_from_hub	…
0	3	5590	…	0	156	
1	2	5856	…	1	79	
2	3	4803	…	1	70	
3	2	4784	…	1	255	
4	4	3699	…	0	205	
…	…	…	…	…	…	
5524	3	5030	…	1	203	
5525	3	4320	…	0	170	
5526	2	5268	…	0	99	
5527	4	4378	…	0	220	
5528	9	3626	…	1	104	

	workers_num	wh_est_year	storage_issue_reported_l3m	temp_reg_mach	…
0	30.0	2006.0	24	0	
1	31.0	2019.0	5	1	
2	41.0	2008.0	19	1	
3	33.0	2017.0	9	1	
4	20.0	1999.0	25	0	
…	…	…	…	…	
5524	28.0	2007.0	22	1	

```
5525         28.0       2010.0                          26              0
5526         22.0       2010.0                          20              0
5527         17.0       2010.0                          13              0
5528         36.0       2010.0                          11              0

      approved_wh_govt_certificate  wh_breakdown_l3m  govt_check_l3m  \
0                                0                 2               5
1                                4                 2              24
2                                1                 5               9
3                                1                 3              11
4                                2                 4              26
…                              …                 …               …
5524                             1                 2               4
5525                             4                 5               9
5526                             3                 6               4
5527                             2                 3              28
5528                             4                 6              26

      product_wg_ton
0              30132
1               6075
2              24076
3              13092
4              29071
…                …
5524           29138
5525           33108
5526           24072
5527           15055
5528           14103

[5529 rows x 24 columns]
```

[37]:
```
#Removing unwanted columns for creating machine learning models.
df3=df2.drop(["Ware_house_ID","WH_Manager_ID"],axis=1)
df3
```

[37]:
```
      Location_type  WH_capacity_size  zone  WH_regional_zone  \
0                 0                 0     1                 4
1                 0                 0     1                 4
2                 0                 2     1                 5
3                 0                 1     3                 3
4                 1                 1     1                 3
…               …                 …   …               …
5524              0                 0     1                 5
5525              0                 2     2                 5
5526              0                 0     1                 5
```

```
5527              0                 0     1                 5
5528              0                 1     0                 3

      num_refill_req_l3m  transport_issue_l1y  Competitor_in_mkt  \
0                      5                    0                  3
1                      5                    0                  2
2                      3                    0                  3
3                      5                    2                  2
4                      6                    0                  4
…                    …                    …                  …
5524                   8                    1                  3
5525                   3                    1                  3
5526                   7                    2                  2
5527                   6                    2                  4
5528                   7                    2                  9

      retail_shop_num  wh_owner_type  distributor_num  …  electric_supply  \
0                5590              0               15  …                0
1                5856              0               40  …                1
2                4803              1               40  …                1
3                4784              1               15  …                1
4                3699              1               24  …                0
…                …               …               …  …              …
5524             5030              1               46  …                1
5525             4320              1               60  …                0
5526             5268              0               64  …                0
5527             4378              1               32  …                0
5528             3626              0               50  …                1

      dist_from_hub  workers_num  wh_est_year  storage_issue_reported_l3m  \
0               156         30.0       2006.0                          24
1                79         31.0       2019.0                           5
2                70         41.0       2008.0                          19
3               255         33.0       2017.0                           9
4               205         20.0       1999.0                          25
…             …           …          …                             …
5524            203         28.0       2007.0                          22
5525            170         28.0       2010.0                          26
5526             99         22.0       2010.0                          20
5527            220         17.0       2010.0                          13
5528            104         36.0       2010.0                          11

      temp_reg_mach  approved_wh_govt_certificate  wh_breakdown_l3m  \
0                 0                             0                 2
1                 1                             4                 2
2                 1                             1                 5
3                 1                             1                 3
```

```
4                  0                                2                4
…                …                                …                …
5524               1                                1                2
5525               0                                4                5
5526               0                                3                6
5527               0                                2                3
5528               0                                4                6

      govt_check_l3m  product_wg_ton
0                  5           30132
1                 24            6075
2                  9           24076
3                 11           13092
4                 26           29071
…               …              …
5524               4           29138
5525               9           33108
5526               4           24072
5527              28           15055
5528              26           14103

[5529 rows x 22 columns]
```

[38]:
```
#Train Test Split
X_test=df3.drop(["product_wg_ton"],axis=1)
X_test.shape
```

[38]:
```
(5529, 21)
```

[39]:
```
y_test=df3["product_wg_ton"]
y_test
```

[39]:
```
0       30132
1        6075
2       24076
3       13092
4       29071
        …
5524    29138
5525    33108
5526    24072
5527    15055
5528    14103
Name: product_wg_ton, Length: 5529, dtype: int64
```

11 Model Selection And Training

[40]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor,␣
 ↪GradientBoostingRegressor,AdaBoostRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
```

12 1.Random Forest Regressor

[41]:
```
rf=RandomForestRegressor()
rf.fit(X_train,y_train)
y_pred=rf.predict(X_test)
y_pred
```

[41]:
```
array([30917.51,  6691.23, 24982.68, …, 23479.18, 15367.89, 13882.67])
```

[43]:
```
#Evaluating model's performance
mse=mean_squared_error(y_pred,y_test)
r2score=r2_score(y_pred,y_test)
msa=mean_absolute_error(y_pred,y_test)
print("mean square error = ",mse)
print("mean absolute error = ",msa)
print("r2 score = ",r2score)
```

```
mean square error =  840047.372569217
mean absolute error =  683.0551745342739
r2 score =  0.9938075839021027
```

13 2.Decision Tree Regressor

[47]:
```
dt=DecisionTreeRegressor()
dt.fit(X_train,y_train)
y_pred=dt.predict(X_test)
y_pred
```

[47]:
```
array([30106.,  7140., 24148., …, 23061., 15135., 14094.])
```

[49]:
```
#Evaluating model's performance
mse=mean_squared_error(y_pred,y_test)
r2score=r2_score(y_pred,y_test)
msa=mean_absolute_error(y_pred,y_test)
print("mean square error = ",mse)
print("mean absolute error = ",msa)
```

```
print("r2 score = ",r2score)
```

```
mean square error =  1562516.4369687105
mean absolute error =  837.0305661059866
r2 score =  0.9885262091399156
```

14 3.Linear Regressor

[50]:
```
Lr=LinearRegression()
Lr.fit(X_train,y_train)
y_pred=Lr.predict(X_test)
y_pred
```

[50]:
```
array([31045.16451053,  8191.04890223, 25053.43125272, …,
       24652.4732326 , 16334.04454806, 13054.1600184 ])
```

[52]:
```
#Evaluating model's performance
mse=mean_squared_error(y_pred,y_test)
r2score=r2_score(y_pred,y_test)
msa=mean_absolute_error(y_pred,y_test)
print("mean square error = ",mse)
print("mean absolute error = ",msa)
print("r2 score = ",r2score)
```

```
mean square error =  3092997.0317738103
mean absolute error =  1303.6993291252022
r2 score =  0.9767368018646007
```

15 4. AddaBoost Regressor

[53]:
```
ada=AdaBoostRegressor()
ada.fit(X_train,y_train)
y_pred=ada.predict(X_test)
y_pred
```

[53]:
```
array([30615.70540383,  7857.59637188, 25229.68610422, …,
       26086.37823372, 17372.12370311, 13570.81484794])
```

[54]:
```
#Evaluating model's performance
mse=mean_squared_error(y_pred,y_test)
msa=mean_absolute_error(y_pred,y_test)
r2score=r2_score(y_pred,y_test)
print("mean square error = ",mse)
print("mean absolute error = ",msa)
print("r2 score = ",r2score)
```

```
mean square error =  3127002.440938234
mean absolute error =  1423.1772833783214
r2 score =  0.9760729418867989
```

16 5.Gradient Boost Regressor

[55]:
```
gb=GradientBoostingRegressor()
gb.fit(X_train,y_train)
y_pred=gb.predict(X_test)
y_pred
```

[55]:
```
array([30966.48678957,  6934.26199558, 25603.561988  , …,
       23554.01466975, 15292.0876425 , 13995.05716719])
```

[57]:
```
#Evaluating model's performance
mse=mean_squared_error(y_pred,y_test)
r2score=r2_score(y_pred,y_test)
msa=mean_absolute_error(y_pred,y_test)
print("mean square error = ",mse)
print("mean absolute error = ",msa)
print("r2 score = ",r2score)
```

```
mean square error =  828181.7936730888
mean absolute error =  699.06139176542
r2 score =  0.9939247277486309
```

17 Conclusion

In conclusion, our machine learning model has made significant strides in optimizing the supply chain for our FMCG company.After thoroughly exploring and evaluating different models including AdaBoost Regressor, Decision Tree Regressor, Random Forest, Linear Regression, and Gradient Boosting Regressor, it has been determined that the Gradient Boosting Regressor model stands out as the most effective choices as the Gradient Boosting Regressor is having the highest r2 score and lowest mean square error and mean absolute error indicating its accurancy in predicting product weights.The second best model is the Random forest regressor.

18 Recommendations

1. it is recommended to use the Random Forest Regressor model into the company's supply chain management system. This will enable real-time decision-making and automatic adjustments in product weight allocations to minimize inventory costs.
2. Manage key features of the model i.e "num_refill_req_l3m","retail_shop_num"and "dist_from_hub." These variables have a significant impact on product weight predictions.

Implementation of these recommendations can lead to cost savings, improved profitability, and more efficient inventory management.

[ ]: