# Business Report - FMCG company

## (Capstone Project)

***by Anbalagan R***

***13-May-2024***

# Contents:

# 1.1) Introduction of the business problem:

**Problem Statement:**

A FMCG company has entered into the instant noodles business two years back. Their higher management has notices that there is a miss match in the demand and supply. Where the demand is high, supply is pretty low and where the demand is low, supply is pretty high. In both the ways it is an inventory cost loss to the company; hence, the higher management wants to optimize the supply quantity in each and every warehouse in entire country.

**Objective:**

* The problem statement is to optimize the supply quantity in each warehouse across the country for a FMCG company's instant noodles business.

* Create a model using historical data to optimize the quantity of products sent to the warehouse and analyze demand patterns across different regions for targeted advertising campaigns.

**Need of the study/project:**

The study/project is needed to address the mismatch between demand and supply observed by the company's management, leading to inventory cost losses.

**Initial Data Limitations:**

At the start, we have access to only limited information. This includes historical data but is not comprehensive. Success with this initial phase will unlock access to more extensive datasets for further analysis and model refinement.

**Progressive Approach:**

We'll begin by demonstrating the effectiveness of our model with the limited data available. Once we prove its impact, the company will provide us with broader access to a comprehensive data lake, allowing us to enhance and expand our model for better decision-making.

**Understanding business/social opportunity:**

The business opportunity lies in building a model using historical data to determine the optimal quantity of product to be shipped to each warehouse, thereby reducing inventory costs and maximizing profits. Additionally, analyzing demand patterns in different regions presents a social opportunity for targeted advertising campaigns, potentially increasing sales and market penetration.

# 1.2) Data Report:

**Understanding how data was collected in terms of time, frequency and methodology:**

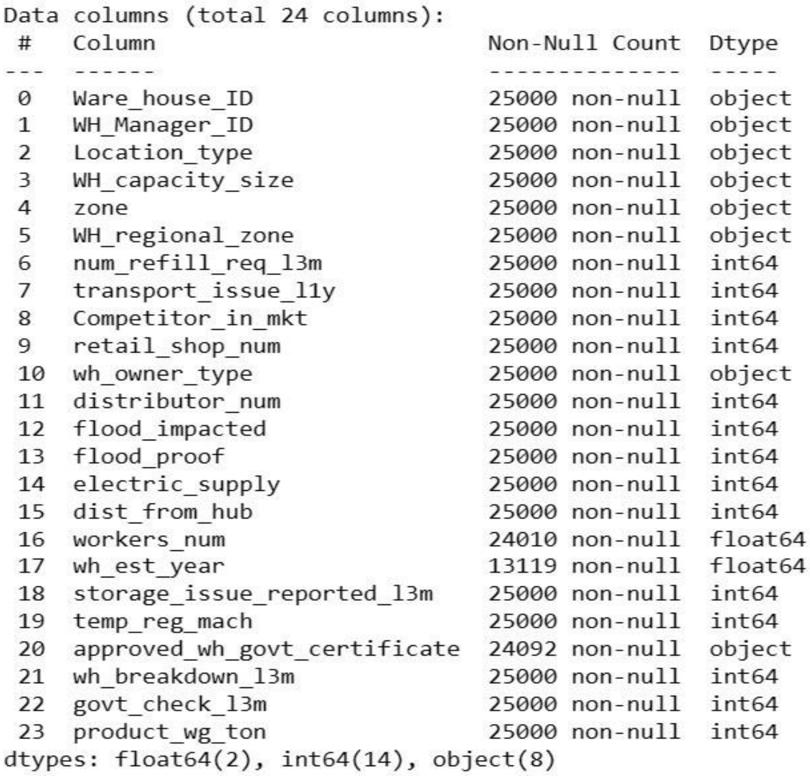
- The company has operated in the instant noodles market for two years, providing a historical context for supply and demand trend analysis. Management recently observed a supply-demand mismatch, prioritizing supply optimization to minimize inventory costs.

- Recent data on product shipments over the last three months, measured in tons, allows for real-time supply quantity analysis, highlighting any fluctuations in shipment volumes.

- Information on warehouse breakdowns in the past three months, including worker strikes, floods, or electrical failures, offers insights into operational challenges. This data helps identify potential disruptions in the supply chain, informing strategies for risk mitigation and operational efficiency improvement.

**Visual inspection of data:**

- The total number of Rows in the Data 25000
- The total number of Columns in the Data 24
- From below table we could identify the datatype of each column also we can see there are few missing values in the data set. Which will be treated in further proceedings.
- There are 2 float64 type , 14 int64 type, 8 object type, data columns in the data set.



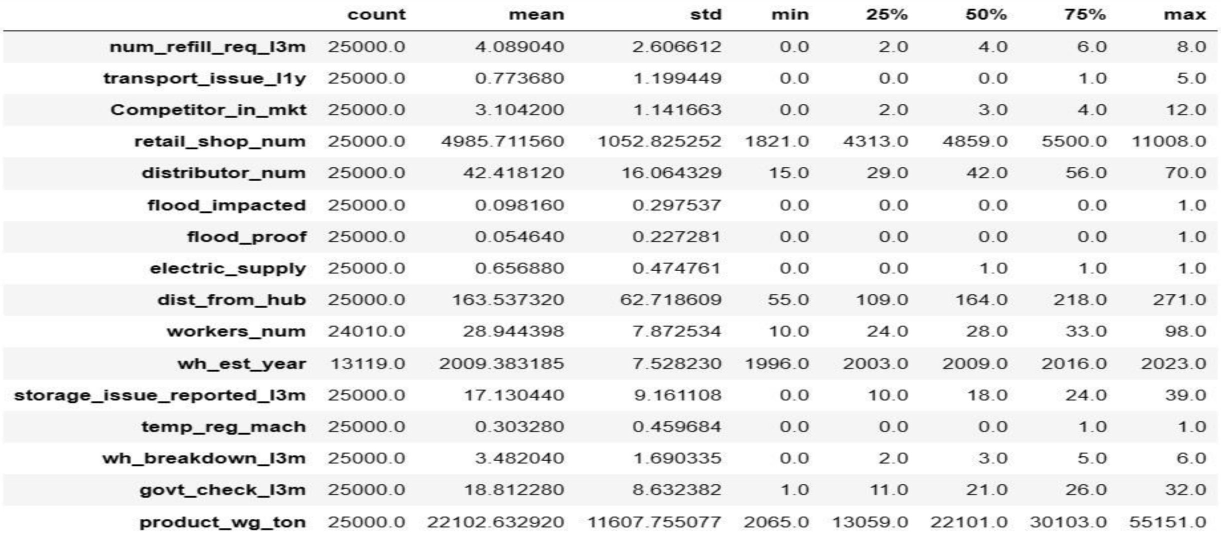
```
Data columns (total 24 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Ware_house_ID                 25000 non-null  object
 1   WH_Manager_ID                 25000 non-null  object
 2   Location_type                 25000 non-null  object
 3   WH_capacity_size              25000 non-null  object
 4   zone                          25000 non-null  object
 5   WH_regional_zone              25000 non-null  object
 6   num_refill_req_l3m            25000 non-null  int64
 7   transport_issue_l1y           25000 non-null  int64
 8   Competitor_in_mkt             25000 non-null  int64
 9   retail_shop_num               25000 non-null  int64
 10  wh_owner_type                 25000 non-null  object
 11  distributor_num               25000 non-null  int64
 12  flood_impacted                25000 non-null  int64
 13  flood_proof                   25000 non-null  int64
 14  electric_supply               25000 non-null  int64
 15  dist_from_hub                 25000 non-null  int64
 16  workers_num                   24010 non-null  float64
 17  wh_est_year                   13119 non-null  float64
 18  storage_issue_reported_l3m    25000 non-null  int64
 19  temp_reg_mach                 25000 non-null  int64
 20  approved_wh_govt_certificate  24092 non-null  object
 21  wh_breakdown_l3m              25000 non-null  int64
 22  govt_check_l3m                25000 non-null  int64
 23  product_wg_ton                25000 non-null  int64
dtypes: float64(2), int64(14), object(8)
```

Table 1.1

**Understanding of attributes :**

- There are no duplicate lines in the data-set.

**Data Summary:**



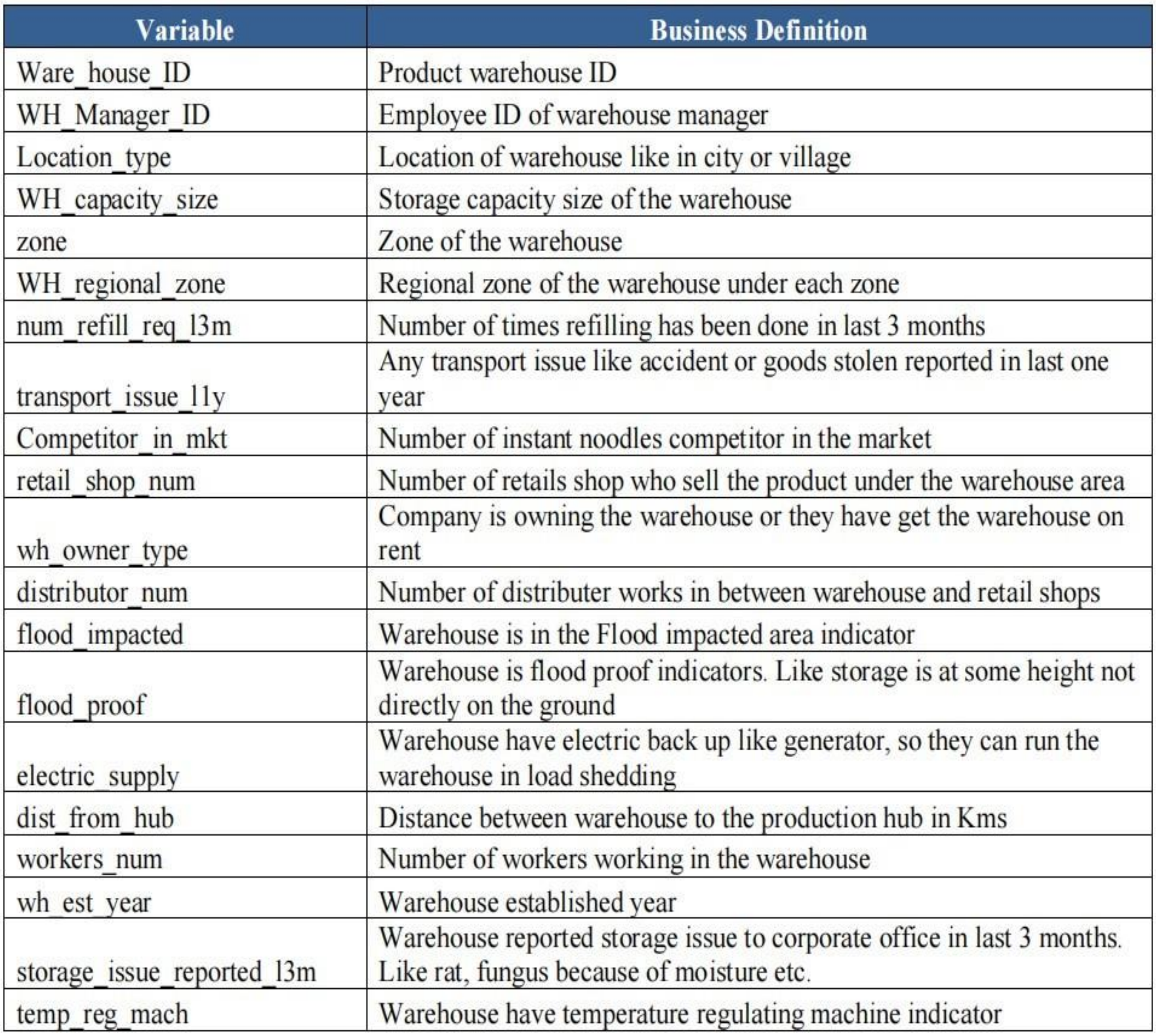
| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| num_refill_req_l3m | 25000.0 | 4.089040 | 2.606612 | 0.0 | 2.0 | 4.0 | 6.0 | 8.0 |
| transport_issue_l1y | 25000.0 | 0.773680 | 1.199449 | 0.0 | 0.0 | 0.0 | 1.0 | 5.0 |
| Competitor_in_mkt | 25000.0 | 3.104200 | 1.141663 | 0.0 | 2.0 | 3.0 | 4.0 | 12.0 |
| retail_shop_num | 25000.0 | 4985.711560 | 1052.825252 | 1821.0 | 4313.0 | 4859.0 | 5500.0 | 11008.0 |
| distributor_num | 25000.0 | 42.418120 | 16.064329 | 15.0 | 29.0 | 42.0 | 56.0 | 70.0 |
| flood_impacted | 25000.0 | 0.098160 | 0.297537 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 |
| flood_proof | 25000.0 | 0.054640 | 0.227281 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 |
| electric_supply | 25000.0 | 0.656880 | 0.474761 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 |
| dist_from_hub | 25000.0 | 163.537320 | 62.718609 | 55.0 | 109.0 | 164.0 | 218.0 | 271.0 |
| workers_num | 24010.0 | 28.944398 | 7.872534 | 10.0 | 24.0 | 28.0 | 33.0 | 98.0 |
| wh_est_year | 13119.0 | 2009.383185 | 7.528230 | 1996.0 | 2003.0 | 2009.0 | 2016.0 | 2023.0 |
| storage_issue_reported_l3m | 25000.0 | 17.130440 | 9.161108 | 0.0 | 10.0 | 18.0 | 24.0 | 39.0 |
| temp_reg_mach | 25000.0 | 0.303280 | 0.459684 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 |
| wh_breakdown_l3m | 25000.0 | 3.482040 | 1.690335 | 0.0 | 2.0 | 3.0 | 5.0 | 6.0 |
| govt_check_l3m | 25000.0 | 18.812280 | 8.632382 | 1.0 | 11.0 | 21.0 | 26.0 | 32.0 |
| product_wg_ton | 25000.0 | 22102.632920 | 11607.755077 | 2065.0 | 13059.0 | 22101.0 | 30103.0 | 55151.0 |

Table 1.2

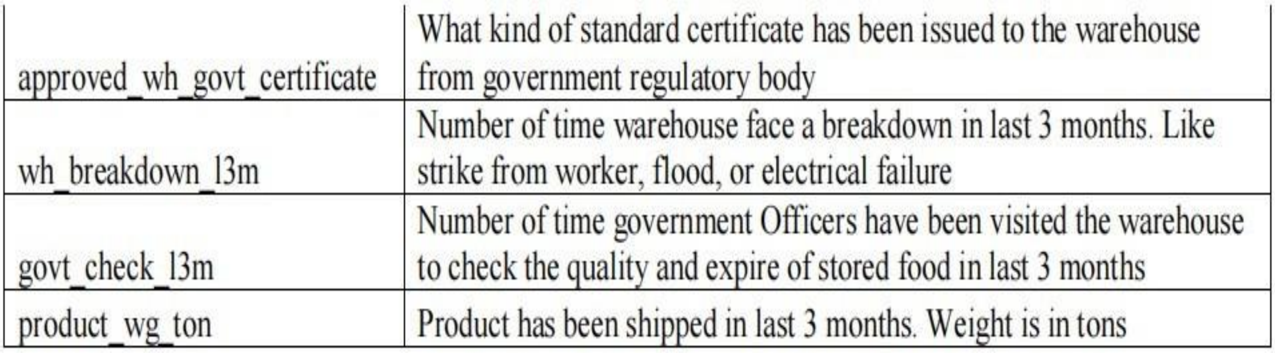
**File**: Data.csv

**Target variable:** product_wg_ton

**Data dictionary:**



| Variable | Business Definition |
|---|---|
| Ware_house_ID | Product warehouse ID |
| WH_Manager_ID | Employee ID of warehouse manager |
| Location_type | Location of warehouse like in city or village |
| WH_capacity_size | Storage capacity size of the warehouse |
| zone | Zone of the warehouse |
| WH_regional_zone | Regional zone of the warehouse under each zone |
| num_refill_req_l3m | Number of times refilling has been done in last 3 months |
| transport_issue_l1y | Any transport issue like accident or goods stolen reported in last one year |
| Competitor_in_mkt | Number of instant noodles competitor in the market |
| retail_shop_num | Number of retails shop who sell the product under the warehouse area |
| wh_owner_type | Company is owning the warehouse or they have get the warehouse on rent |
| distributor_num | Number of distributer works in between warehouse and retail shops |
| flood_impacted | Warehouse is in the Flood impacted area indicator |
| flood_proof | Warehouse is flood proof indicators. Like storage is at some height not directly on the ground |
| electric_supply | Warehouse have electric back up like generator, so they can run the warehouse in load shedding |
| dist_from_hub | Distance between warehouse to the production hub in Kms |
| workers_num | Number of workers working in the warehouse |
| wh_est_year | Warehouse established year |
| storage_issue_reported_l3m | Warehouse reported storage issue to corporate office in last 3 months. Like rat, fungus because of moisture etc. |
| temp_reg_mach | Warehouse have temperature regulating machine indicator |



| | |
|---|---|
| approved_wh_govt_certificate | What kind of standard certificate has been issued to the warehouse from government regulatory body |
| wh_breakdown_l3m | Number of time warehouse face a breakdown in last 3 months. Like strike from worker, flood, or electrical failure |
| govt_check_l3m | Number of time government Officers have been visited the warehouse to check the quality and expire of stored food in last 3 months |
| product_wg_ton | Product has been shipped in last 3 months. Weight is in tons |

Table 1.3

- Data from the past few years has been supplied for analysis.
- We've accounted for 4 main zones and 6 regional zones, along with their respective locations.
- The data-set includes analysis on the impact of natural calamities like floods on the area.
- Information on storage capacity, electricity availability, and temperature conditions has been gathered.
- Details regarding the total number of workers, manager IDs, and warehouse IDs have been provided.
- Competitor counts have been compiled alongside details on government-issued certificates.
- Additionally, some columns include data from the last three months.

# 1.3) Exploratory data analysis

- **Uni-variate analysis:**

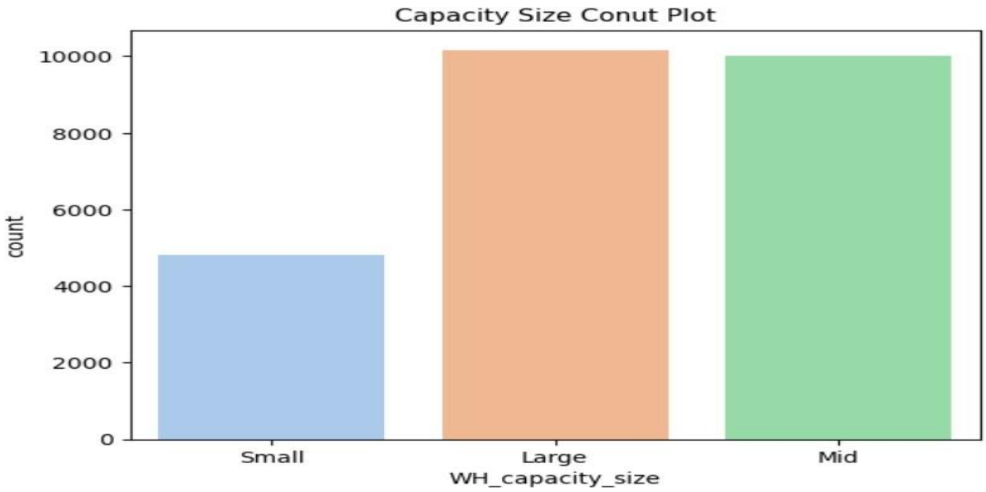


Fig 1.1

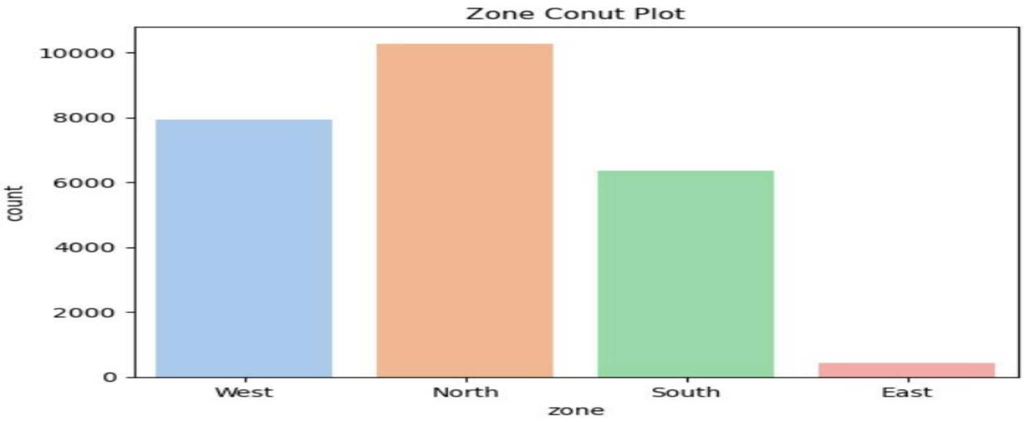


Fig 1.2

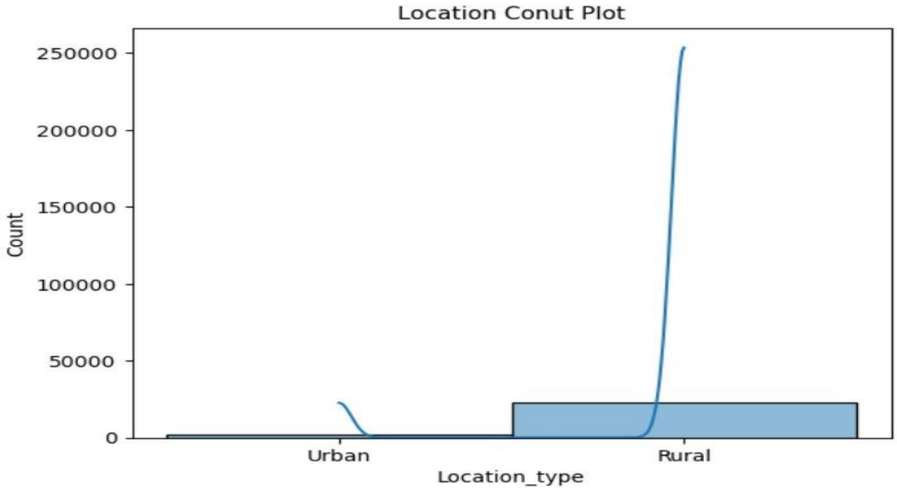


Fig 1.3

- Highest shipment is done in the Northern Region.
- Large capacity and Medium capacity outlets are similar in counts.
- East Region has the lowest shipment recorded. South region is the second lowest.

- **Bi-variate analysis:**

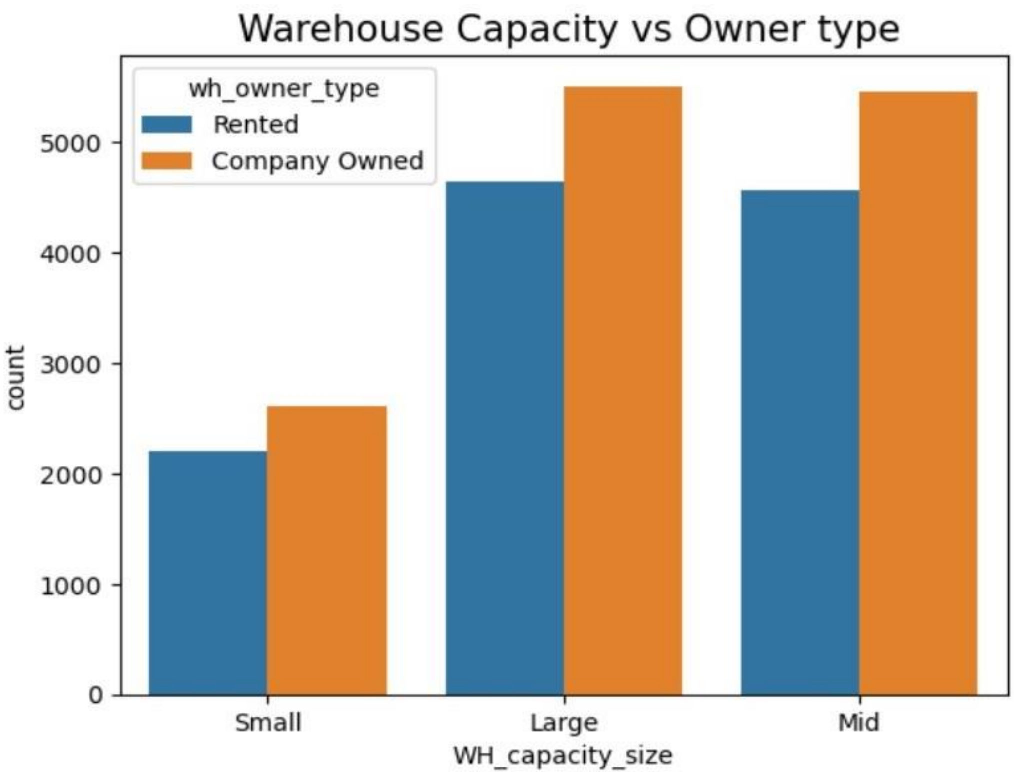


Fig 1.4

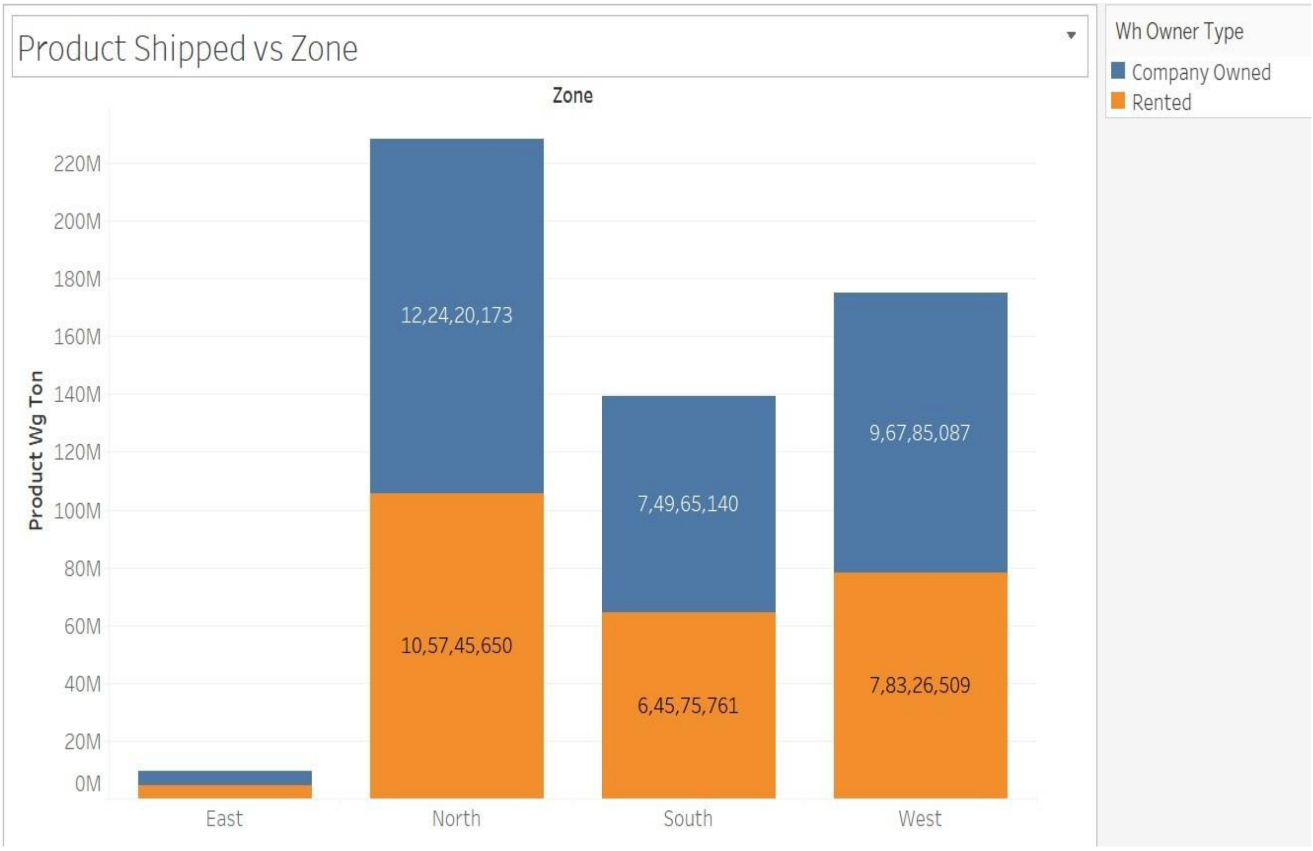


Fig 1.5

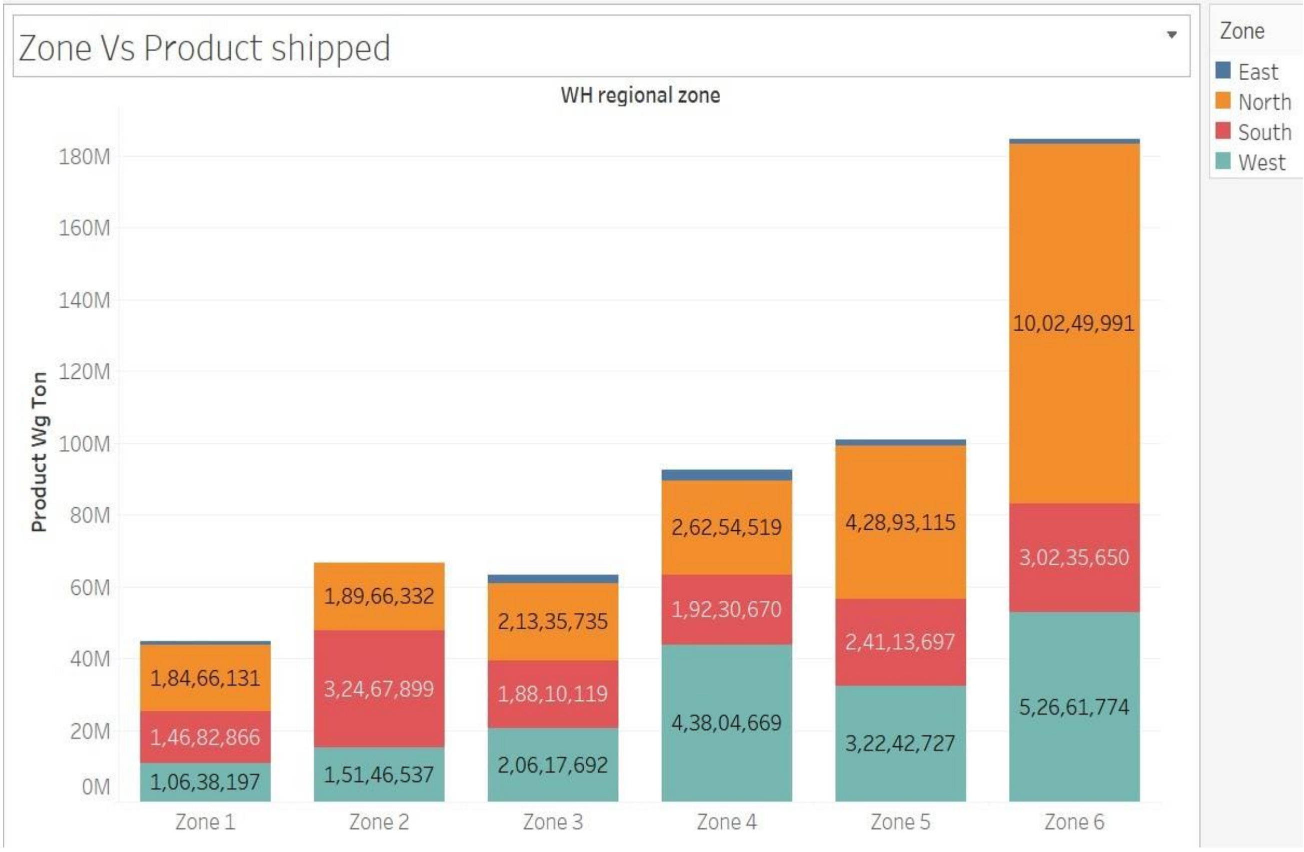

Zone Vs Product shipped
WH regional zone
Product Wg Ton
180M
160M
140M
120M
100M
80M
60M
40M
20M
0M
Zone 1
Zone 2
Zone 3
Zone 4
Zone 5
Zone 6
1,84,66,131
1,46,82,866
1,06,38,197
1,89,66,332
3,24,67,899
1,51,46,537
2,13,35,735
1,88,10,119
2,06,17,692
2,62,54,519
1,92,30,670
4,38,04,669
4,28,93,115
2,41,13,697
3,22,42,727
10,02,49,991
3,02,35,650
5,26,61,774
Zone
East
North
South
West

Fig 1.6

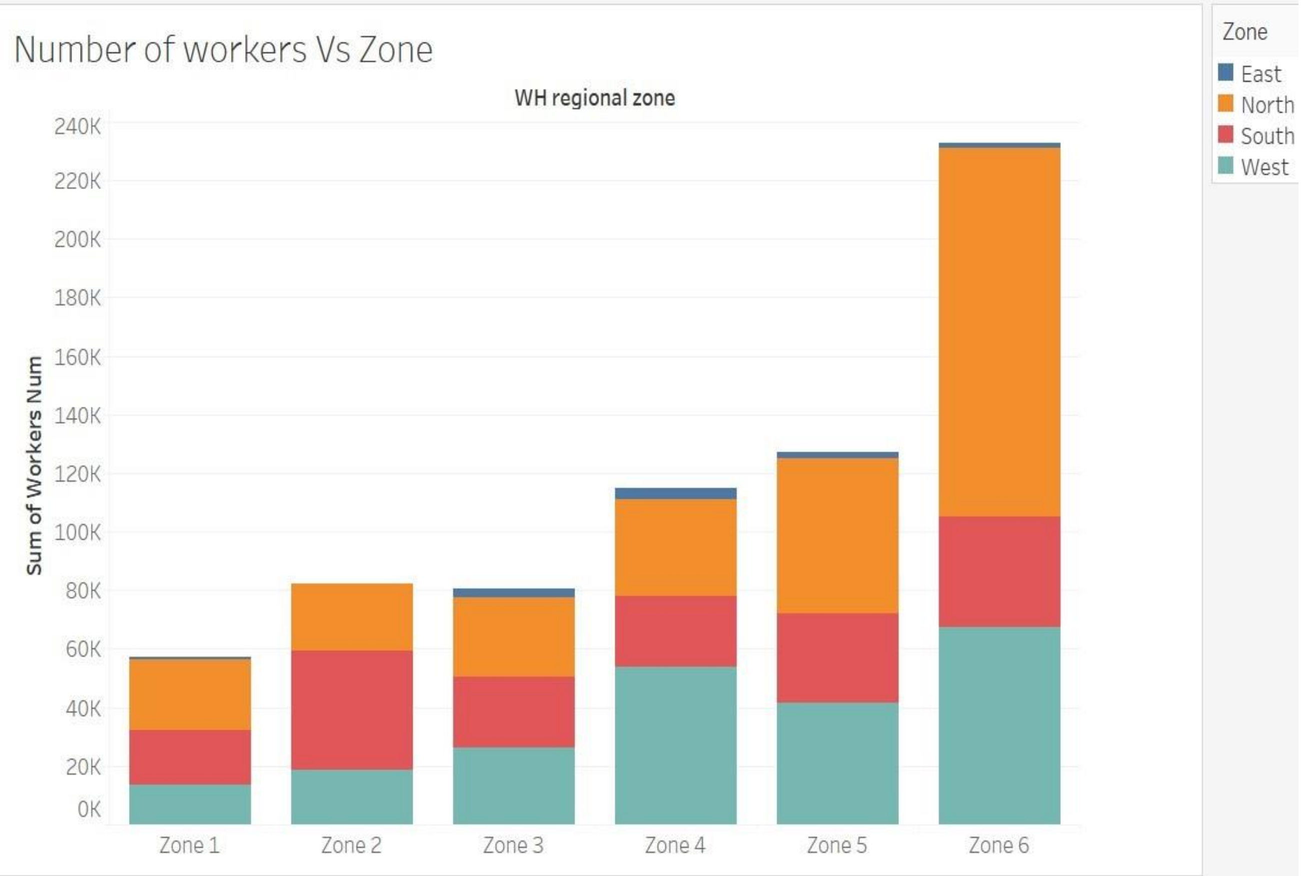

Number of workers Vs Zone
WH regional zone
Sum of Workers Num
240K
220K
200K
180K
160K
140K
120K
100K
80K
60K
40K
20K
0K
Zone 1
Zone 2
Zone 3
Zone 4
Zone 5
Zone 6
Zone
East
North
South
West

Fig 1.7

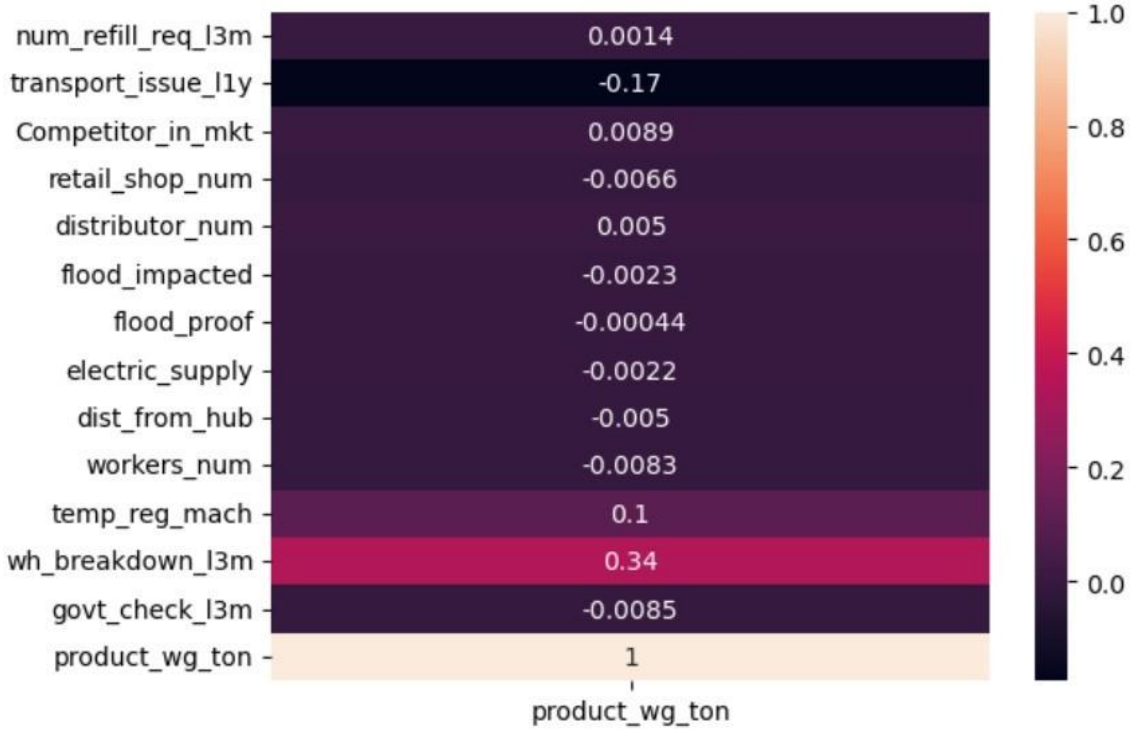


Fig 1.8

The target variable “product_wg_ton” has the highest correlation with “wh_breakdonw_l13m”,”transport_issue_l1y” and “temp_reg_mach” respectively.

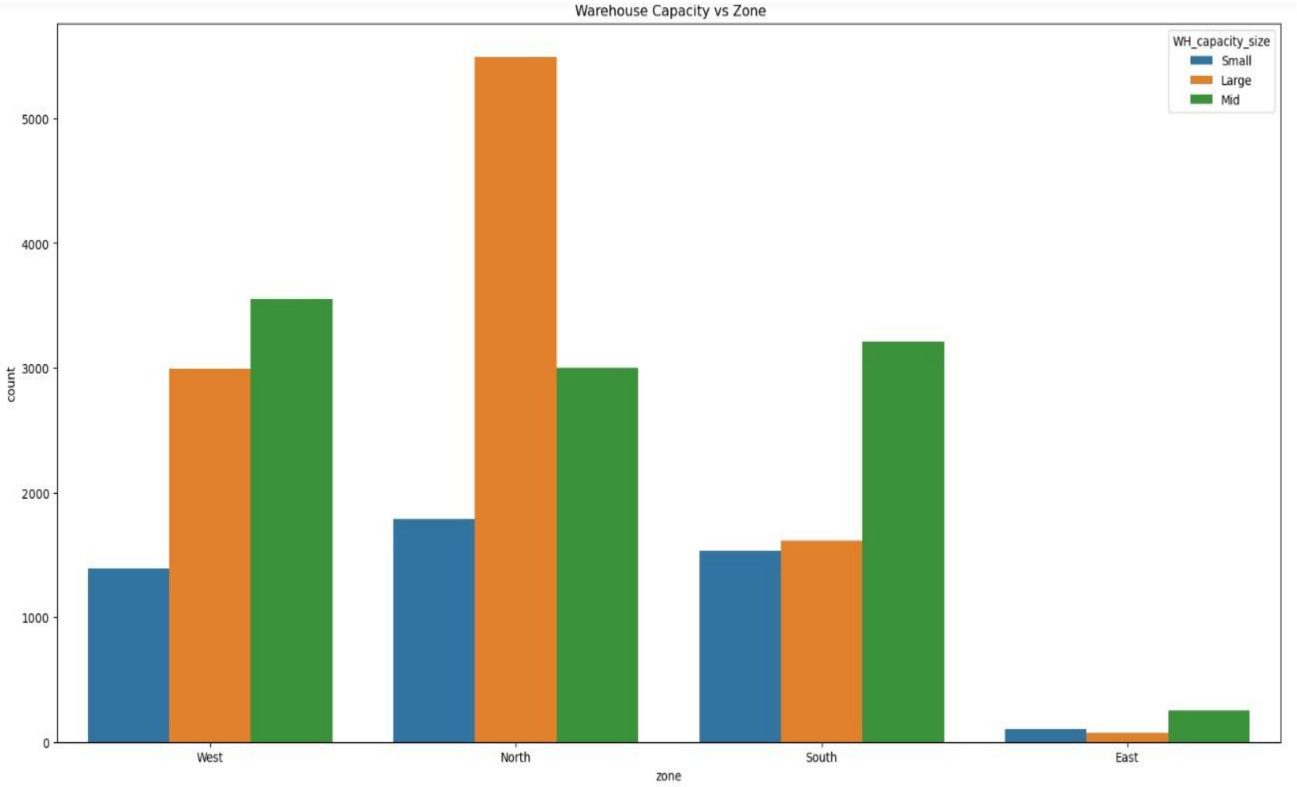


Fig 1.9

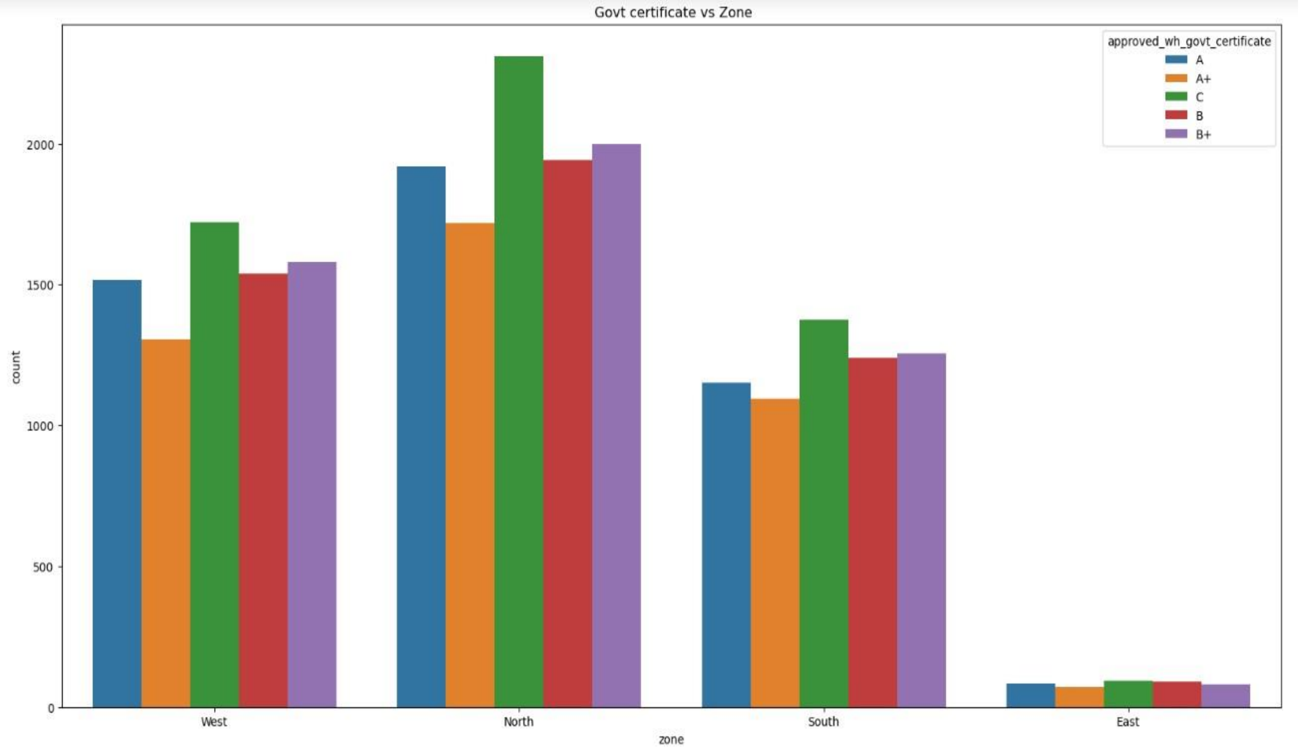


Fig 1.10

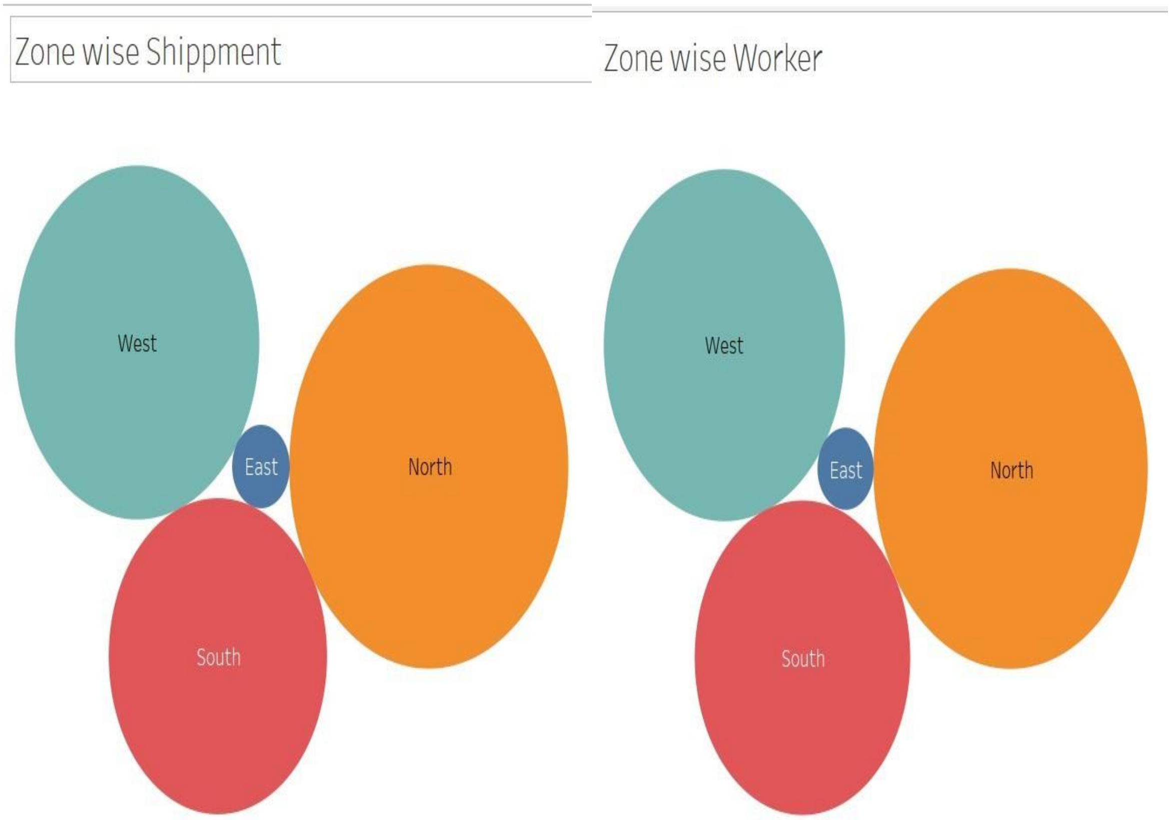


Fig 1.11

Based above chart we can conclude that, the number of workers are precisely/effectively used with respect to the product movement. No excess man power is wasted in any of the region.
In west region we can slightly reduce the man power. Which is slighted higher than the required resource.

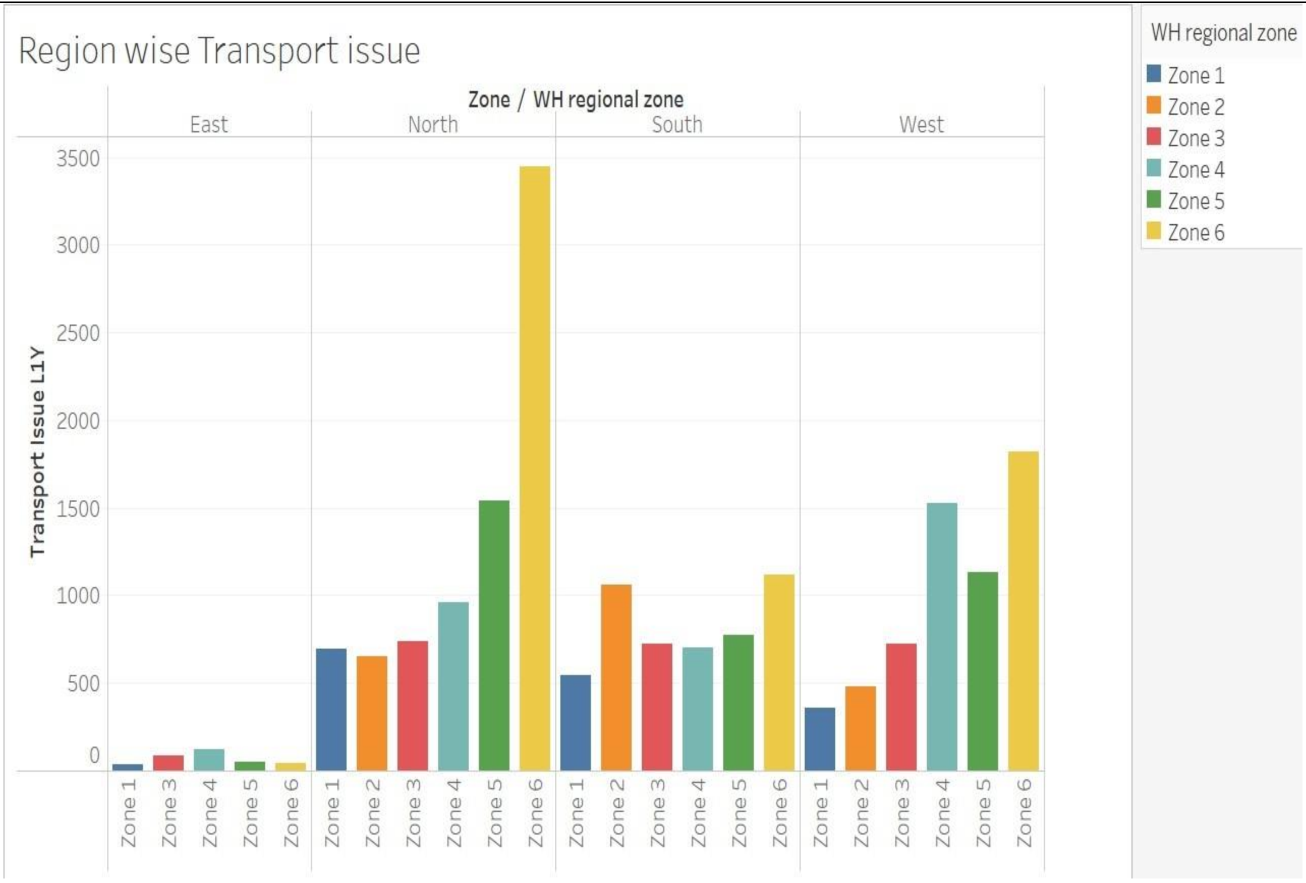


Fig 1.12

Except the East Region all other regions are facing transport issue in zone 6.

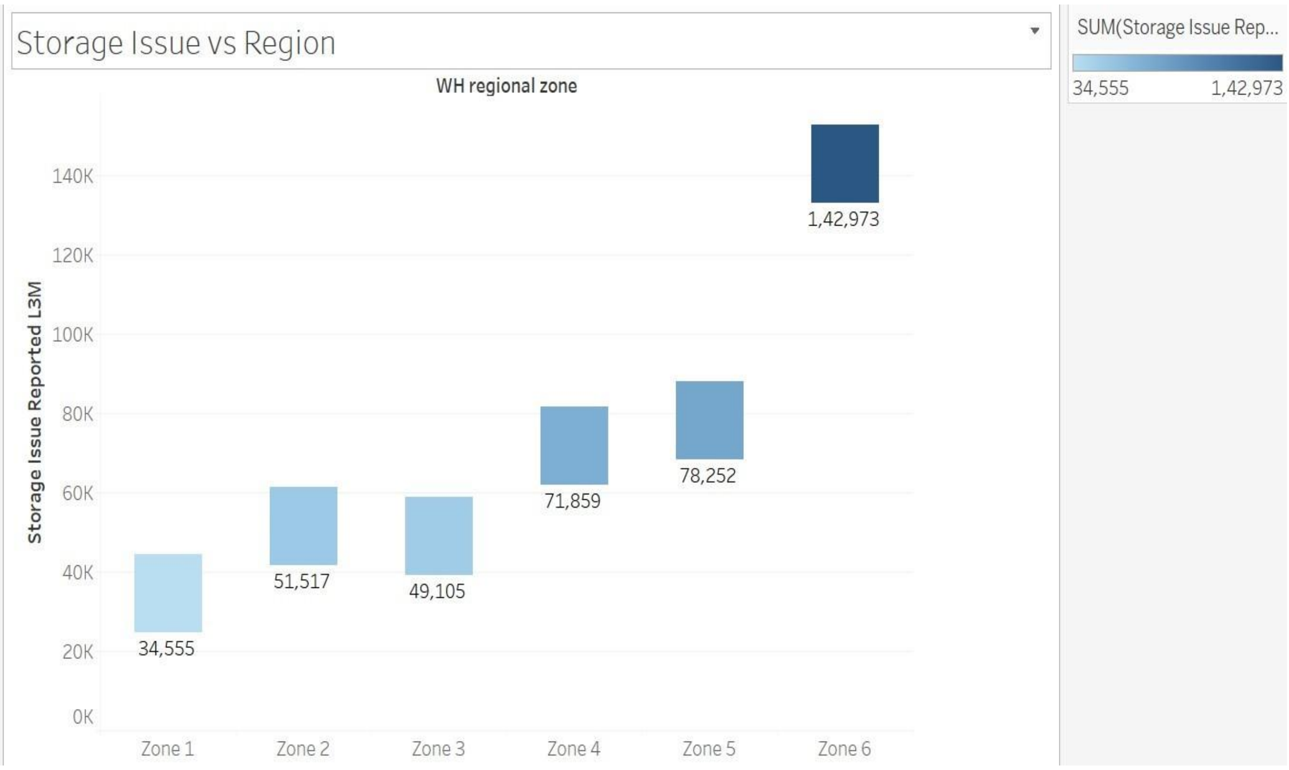


Fig 1.13

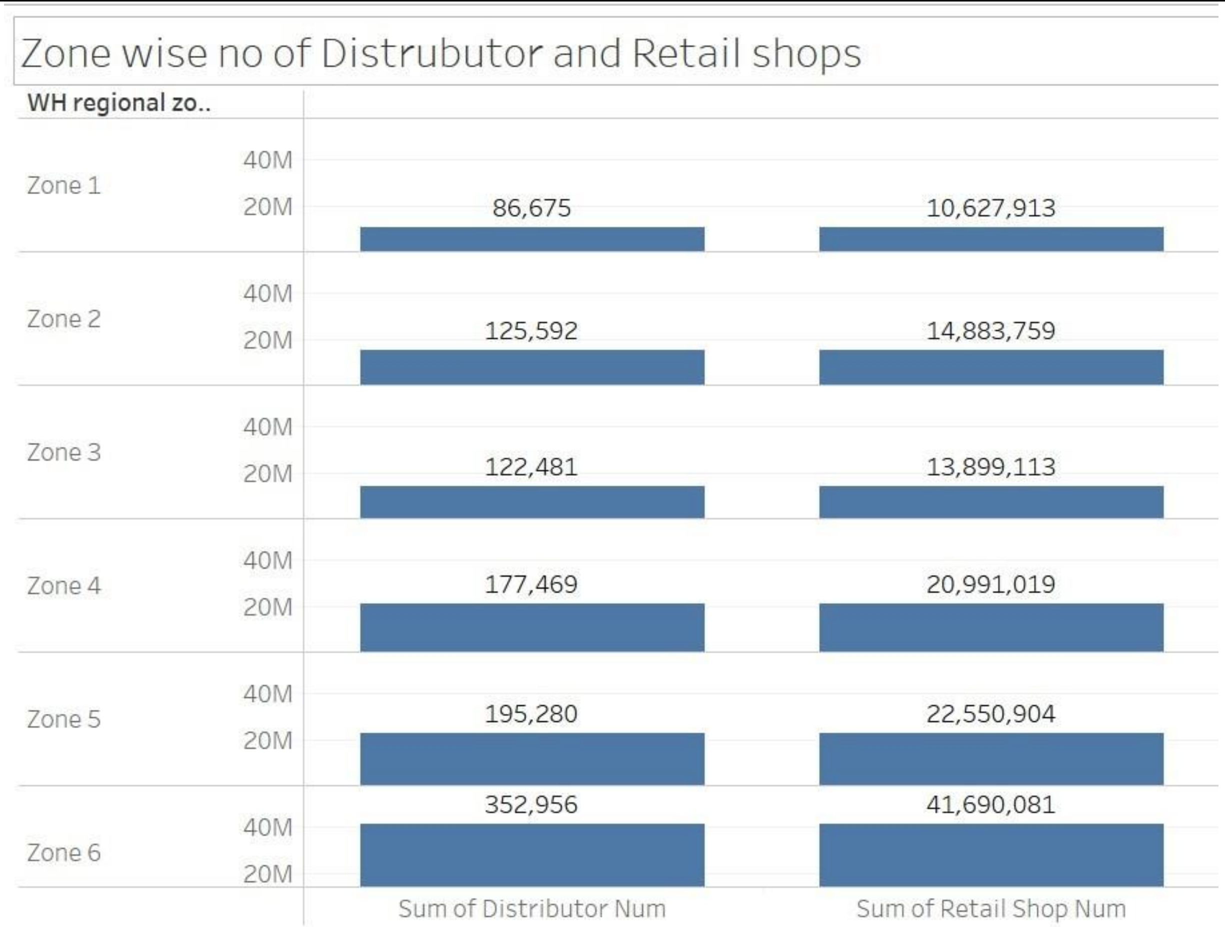
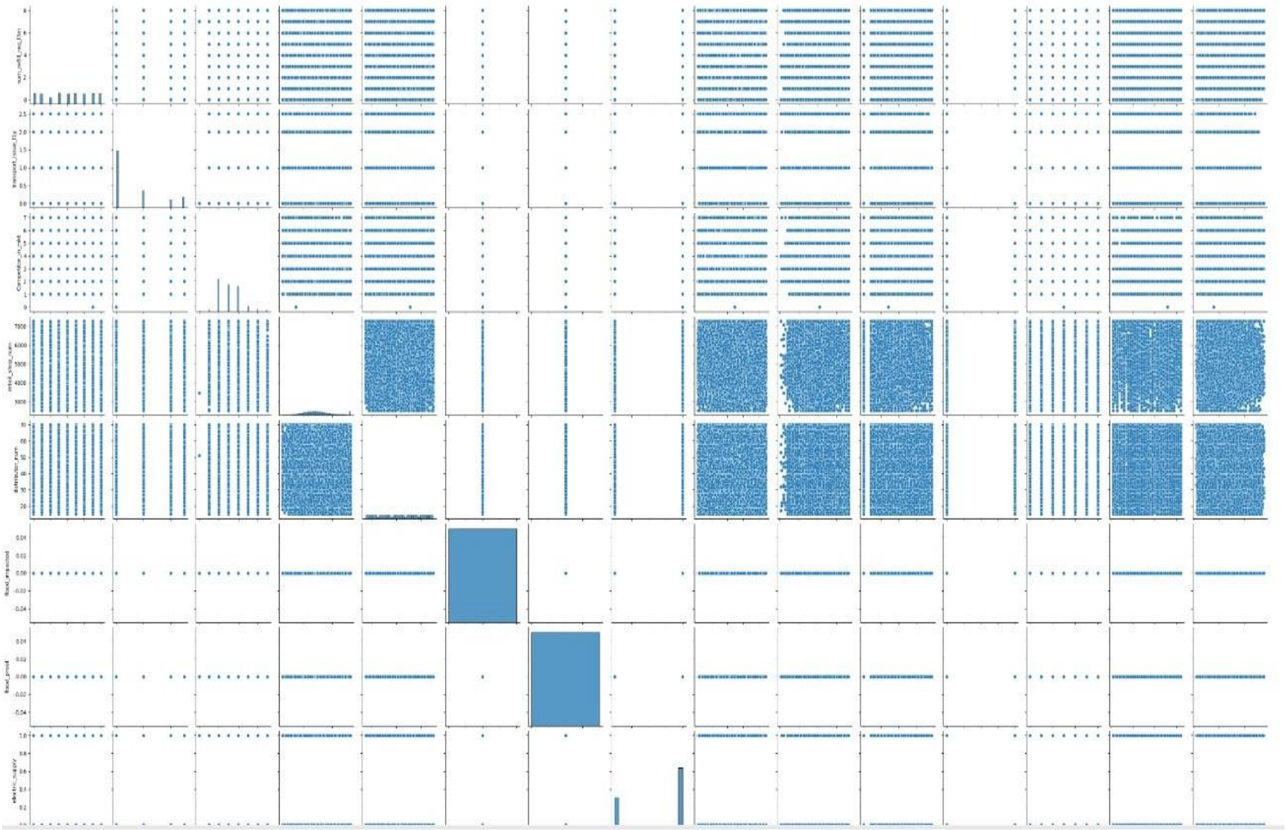


Table 1.4

Pairplot:



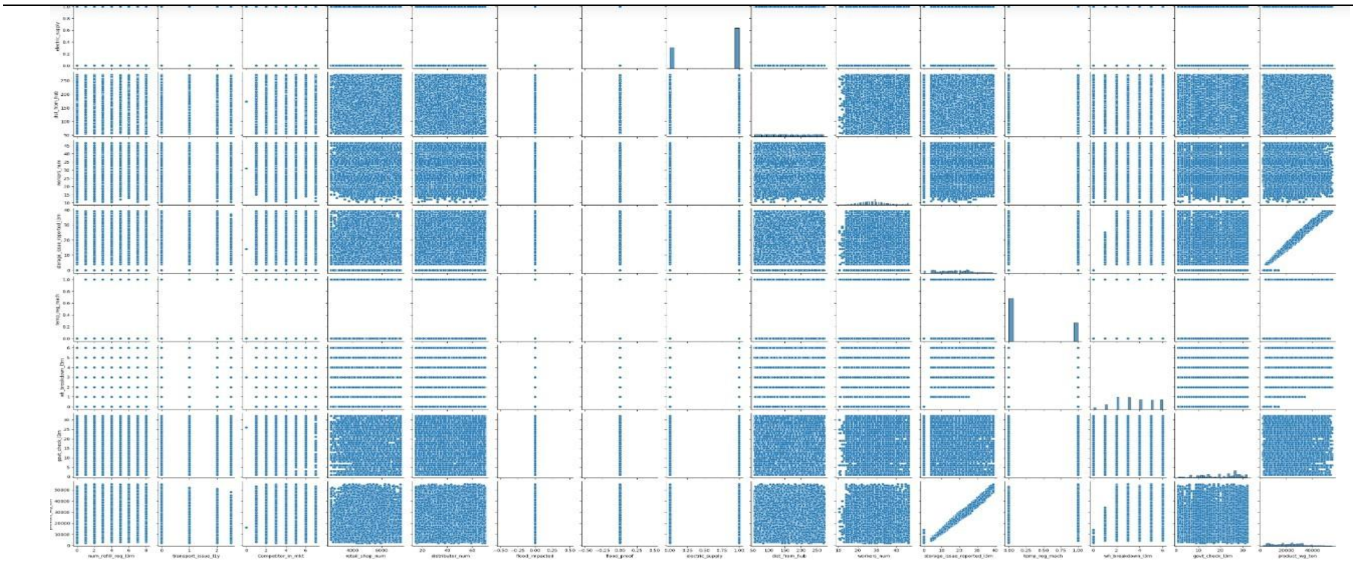


Fig 1.14

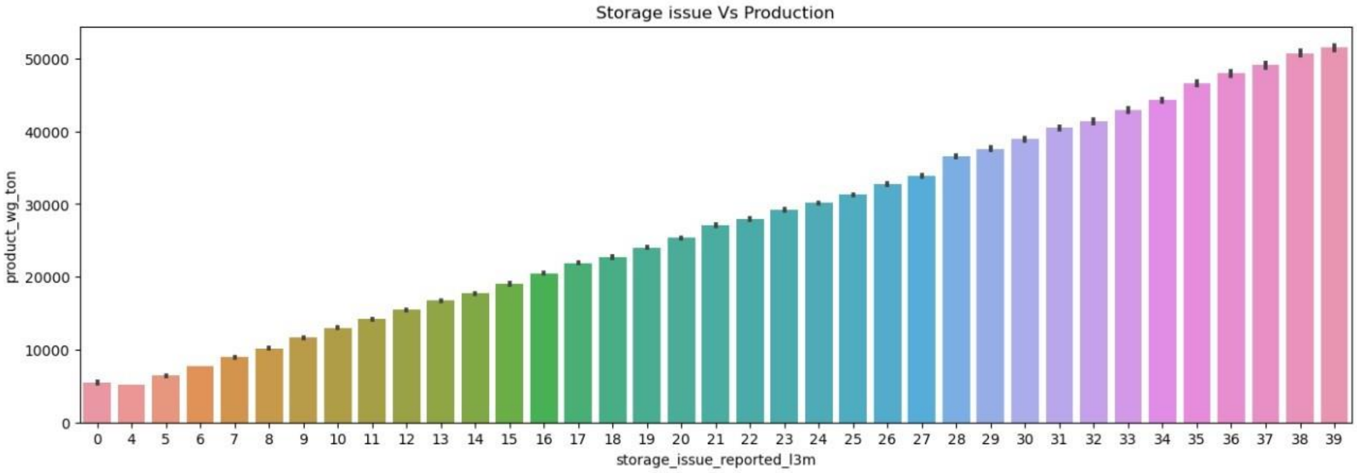


Fig 1.14

Higher the production higher the storage issue.

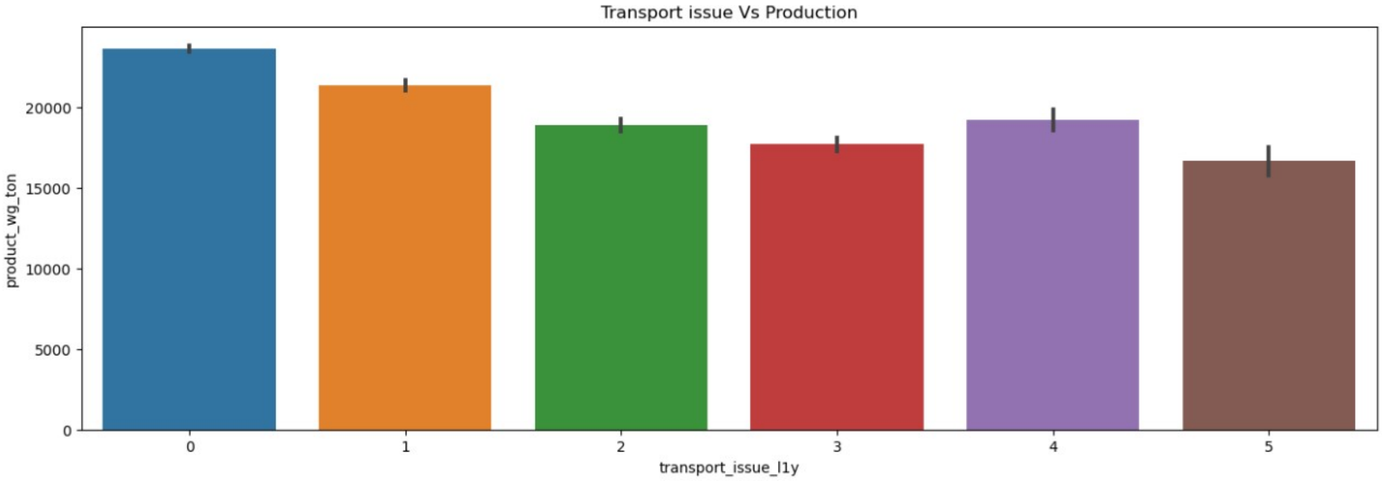


Fig 1.15

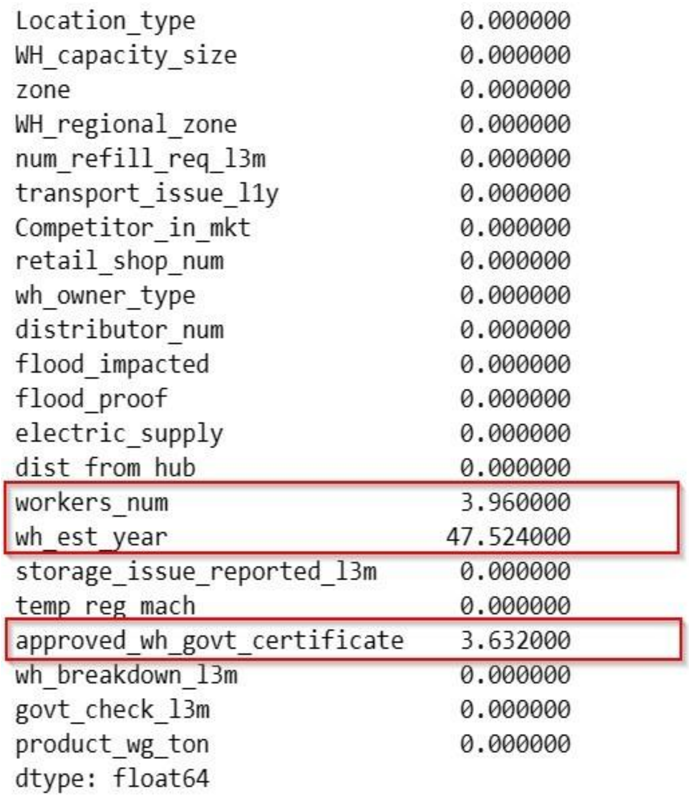
## 1.4) Data Cleaning and Pre-processing.

□ **Removal of unwanted variables:**

Removing columns 'Ware_house_ID' and 'WH_Manager_ID, since they don't add any value to the analysis.

□ **Missing Value treatment:**

From below table we can see that, we have missing values in three columns - Total percentage of null values in the data-set : 2.2965% (~2%).



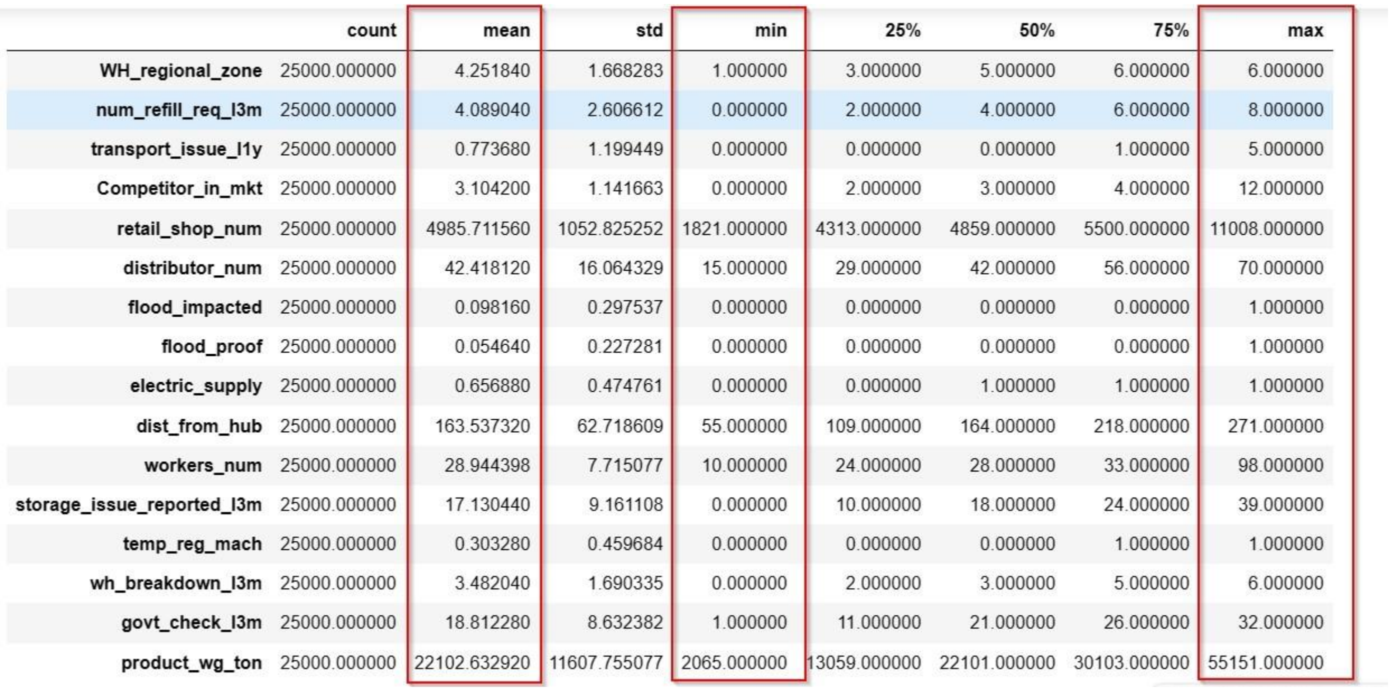
```
Location_type                   0.000000
WH_capacity_size                0.000000
zone                            0.000000
WH_regional_zone                0.000000
num_refill_req_l3m              0.000000
transport_issue_l1y             0.000000
Competitor_in_mkt               0.000000
retail_shop_num                 0.000000
wh_owner_type                   0.000000
distributor_num                 0.000000
flood_impacted                  0.000000
flood_proof                     0.000000
electric_supply                 0.000000
dist_from_hub                   0.000000
workers_num                     3.960000
wh_est_year                    47.524000
storage_issue_reported_l3m      0.000000
temp_reg_mach                   0.000000
approved_wh_govt_certificate    3.632000
wh_breakdown_l3m                0.000000
govt_check_l3m                  0.000000
product_wg_ton                  0.000000
dtype: float64
```

Table 1.5

- It is impossible to have a warehouse with '0' worker numbers, also it is a essential data for model building, so treating the missing value with "mean".
- Warehouse estimated year is not required to build the model, also it has more than 30% missing value so dropping the column. ("wh_est_year", missing ~48%)
- Govt approval certificate is a considerable feature to build a model so we can treat it with mode.
- Total percentage of null values in the data-set : 2.2965% (~2%).

□ **Outlier treatment:**

Outliers were within the range a countable whole number. So keeping the data as it is.



| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **WH_regional_zone** | 25000.000000 | 4.251840 | 1.668283 | 1.000000 | 3.000000 | 5.000000 | 6.000000 | 6.000000 |
| **num_refill_req_l3m** | 25000.000000 | 4.089040 | 2.606612 | 0.000000 | 2.000000 | 4.000000 | 6.000000 | 8.000000 |
| **transport_issue_l1y** | 25000.000000 | 0.773680 | 1.199449 | 0.000000 | 0.000000 | 0.000000 | 1.000000 | 5.000000 |
| **Competitor_in_mkt** | 25000.000000 | 3.104200 | 1.141663 | 0.000000 | 2.000000 | 3.000000 | 4.000000 | 12.000000 |
| **retail_shop_num** | 25000.000000 | 4985.711560 | 1052.825252 | 1821.000000 | 4313.000000 | 4859.000000 | 5500.000000 | 11008.000000 |
| **distributor_num** | 25000.000000 | 42.418120 | 16.064329 | 15.000000 | 29.000000 | 42.000000 | 56.000000 | 70.000000 |
| **flood_impacted** | 25000.000000 | 0.098160 | 0.297537 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 1.000000 |
| **flood_proof** | 25000.000000 | 0.054640 | 0.227281 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 1.000000 |
| **electric_supply** | 25000.000000 | 0.656880 | 0.474761 | 0.000000 | 0.000000 | 1.000000 | 1.000000 | 1.000000 |
| **dist_from_hub** | 25000.000000 | 163.537320 | 62.718609 | 55.000000 | 109.000000 | 164.000000 | 218.000000 | 271.000000 |
| **workers_num** | 25000.000000 | 28.944398 | 7.715077 | 10.000000 | 24.000000 | 28.000000 | 33.000000 | 98.000000 |
| **storage_issue_reported_l3m** | 25000.000000 | 17.130440 | 9.161108 | 0.000000 | 10.000000 | 18.000000 | 24.000000 | 39.000000 |
| **temp_reg_mach** | 25000.000000 | 0.303280 | 0.459684 | 0.000000 | 0.000000 | 0.000000 | 1.000000 | 1.000000 |
| **wh_breakdown_l3m** | 25000.000000 | 3.482040 | 1.690335 | 0.000000 | 2.000000 | 3.000000 | 5.000000 | 6.000000 |
| **govt_check_l3m** | 25000.000000 | 18.812280 | 8.632382 | 1.000000 | 11.000000 | 21.000000 | 26.000000 | 32.000000 |
| **product_wg_ton** | 25000.000000 | 22102.632920 | 11607.755077 | 2065.000000 | 13059.000000 | 22101.000000 | 30103.000000 | 55151.000000 |

Table 1.6 (Describe Data)

□ **Duplicate:**

After removing the unique values, it has been determined that there are **no duplicate** lines present in the data-set.

Note : Before checking for duplicate values, the unique features, such as ID numbers, are removed from the data-set. Otherwise, duplicates will not be identified.

□ **Encoding :**

Transformed following discrete categorical variables into binary vectors using **one-hot encoding**, preparing it for further analysis or modeling

Discrete categorical variables/columns in the data-set

- Location_type
- WH_capacity_size
- zone
- wh_owner_type
- approved_wh_govt_certificate

# 1.5) Model building.

**Clustering:**

**Elbow plot:**

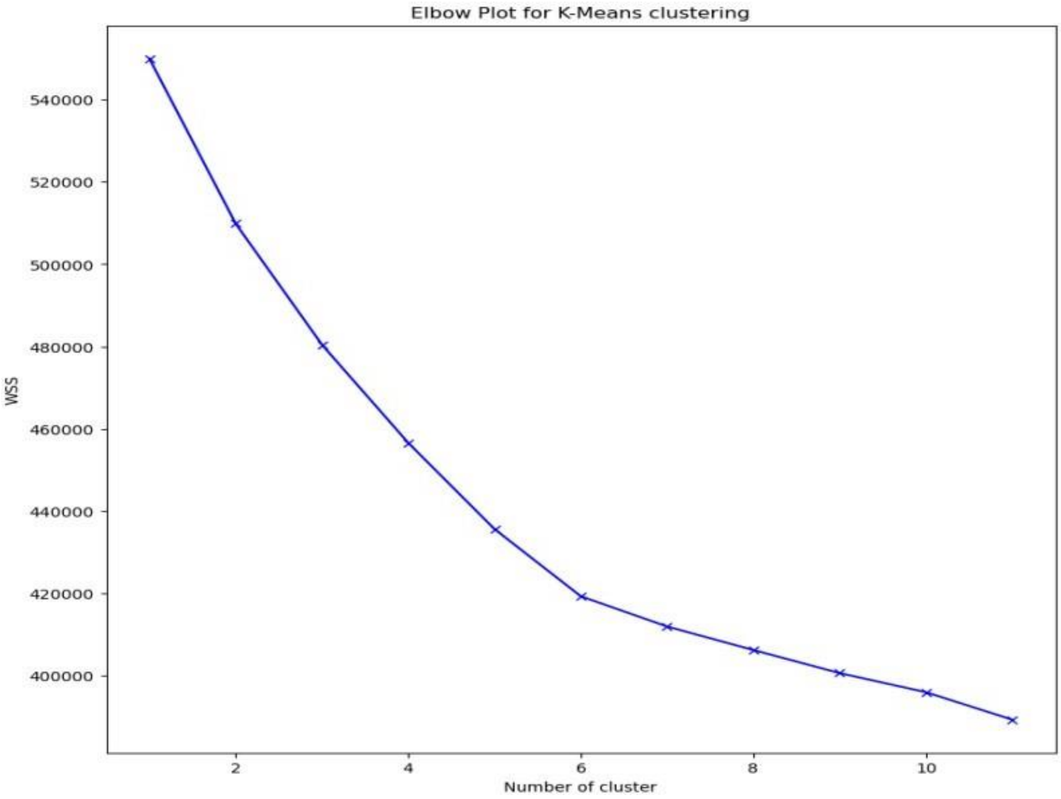


Fig 1.16 : Elbow plot of k-means cluster

**Schilloute**

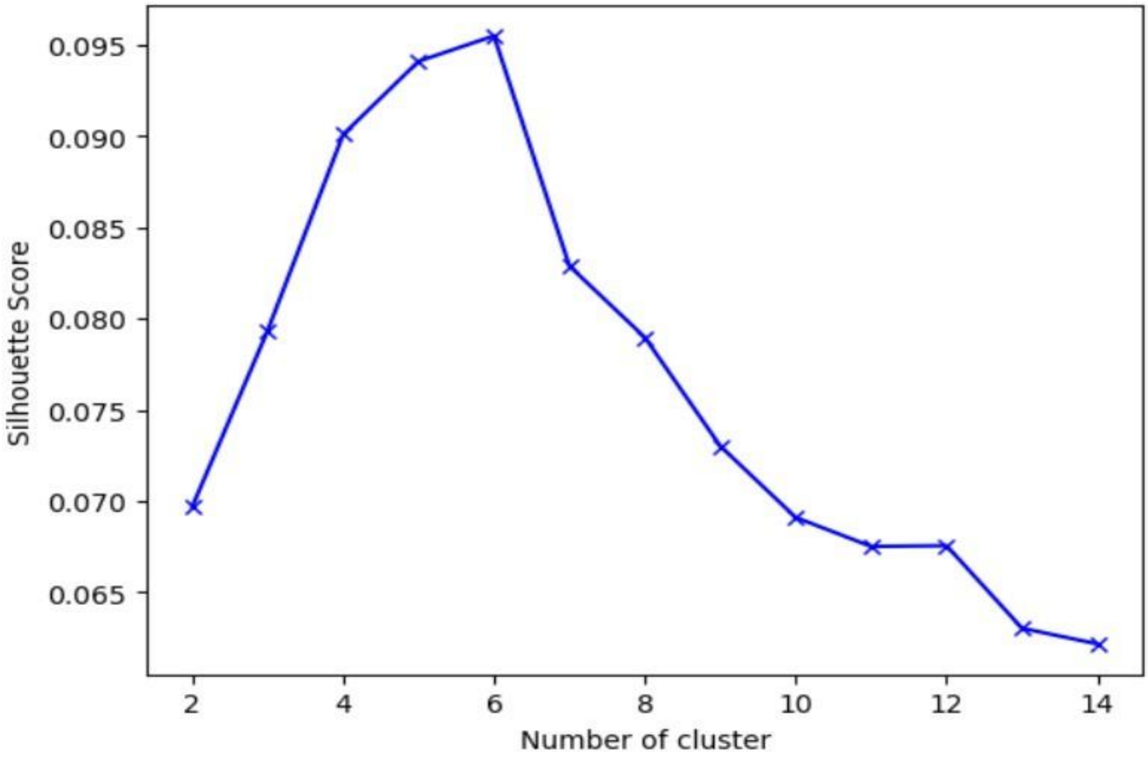


Fig 1.17

- The elbow plot above indicates a sharp decrease in inertia at 6 clusters, suggesting that 6 is the optimal number of clusters.

Dendogram

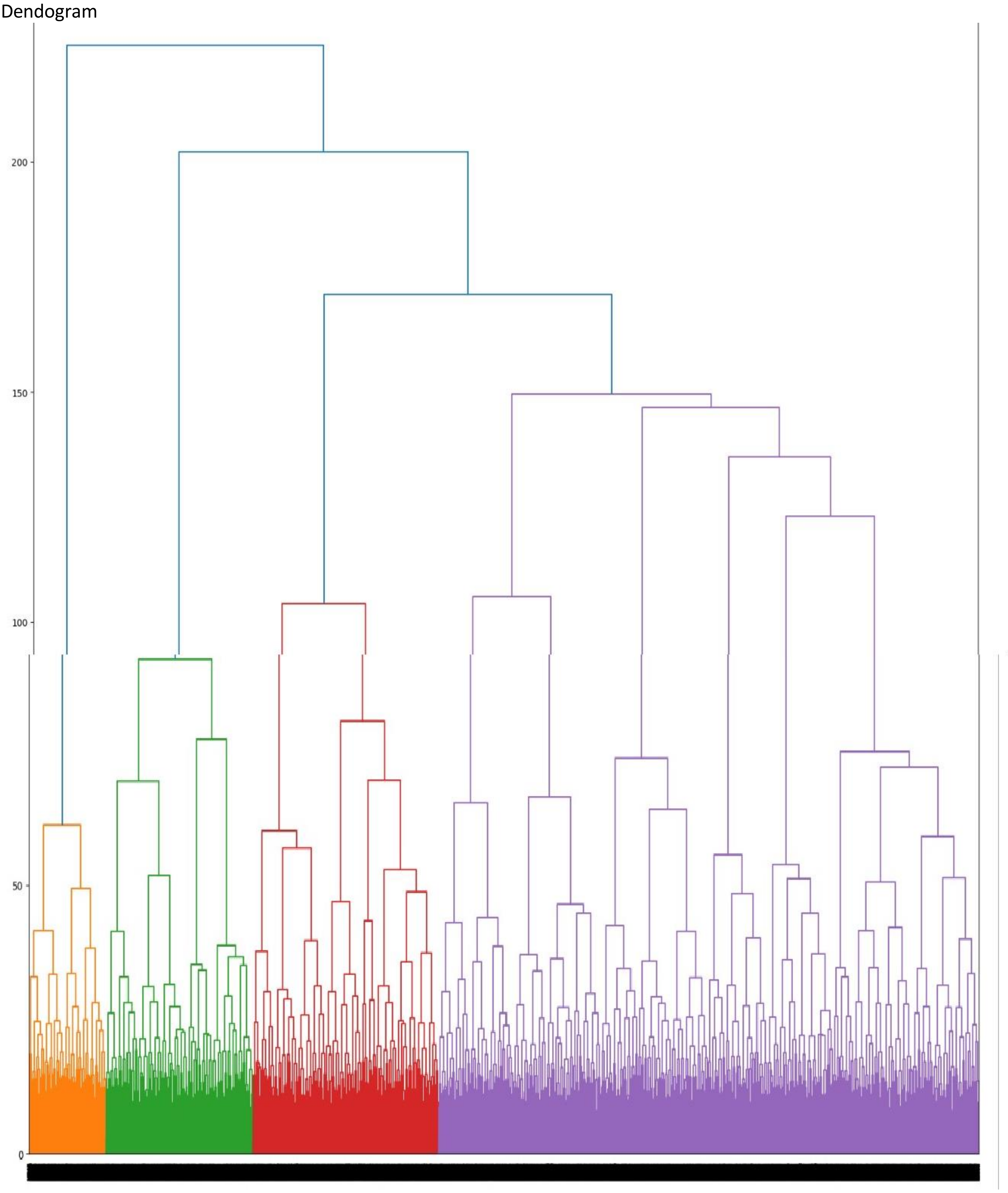
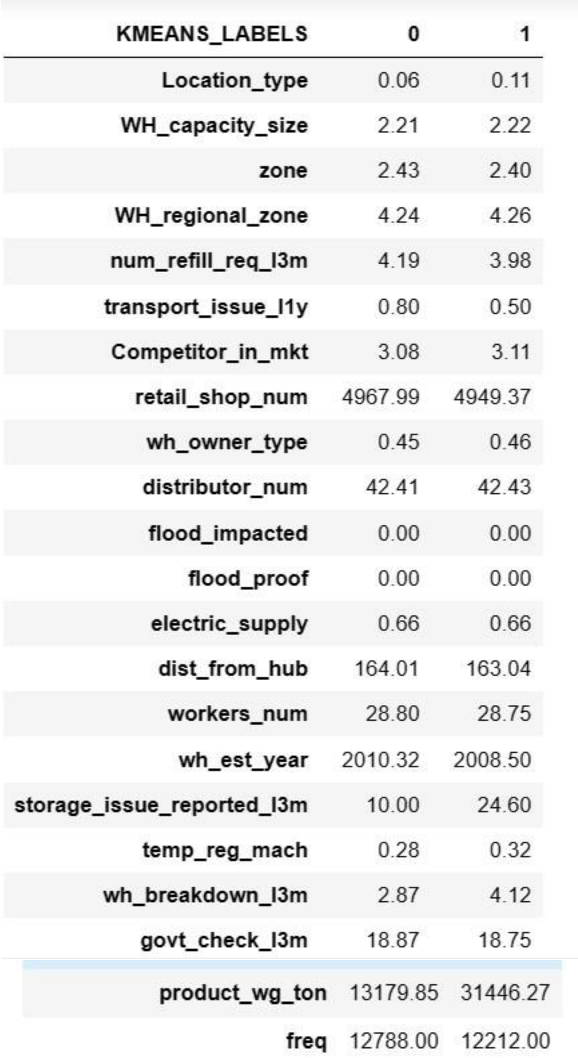


Fig 1.18

Clustering : 6 is the optimal number of clusters.



| KMEANS_LABELS | 0 | 1 |
|---|---|---|
| Location_type | 0.06 | 0.11 |
| WH_capacity_size | 2.21 | 2.22 |
| zone | 2.43 | 2.40 |
| WH_regional_zone | 4.24 | 4.26 |
| num_refill_req_l3m | 4.19 | 3.98 |
| transport_issue_l1y | 0.80 | 0.50 |
| Competitor_in_mkt | 3.08 | 3.11 |
| retail_shop_num | 4967.99 | 4949.37 |
| wh_owner_type | 0.45 | 0.46 |
| distributor_num | 42.41 | 42.43 |
| flood_impacted | 0.00 | 0.00 |
| flood_proof | 0.00 | 0.00 |
| electric_supply | 0.66 | 0.66 |
| dist_from_hub | 164.01 | 163.04 |
| workers_num | 28.80 | 28.75 |
| wh_est_year | 2010.32 | 2008.50 |
| storage_issue_reported_l3m | 10.00 | 24.60 |
| temp_reg_mach | 0.28 | 0.32 |
| wh_breakdown_l3m | 2.87 | 4.12 |
| govt_check_l3m | 18.87 | 18.75 |
| product_wg_ton | 13179.85 | 31446.27 |
| freq | 12788.00 | 12212.00 |

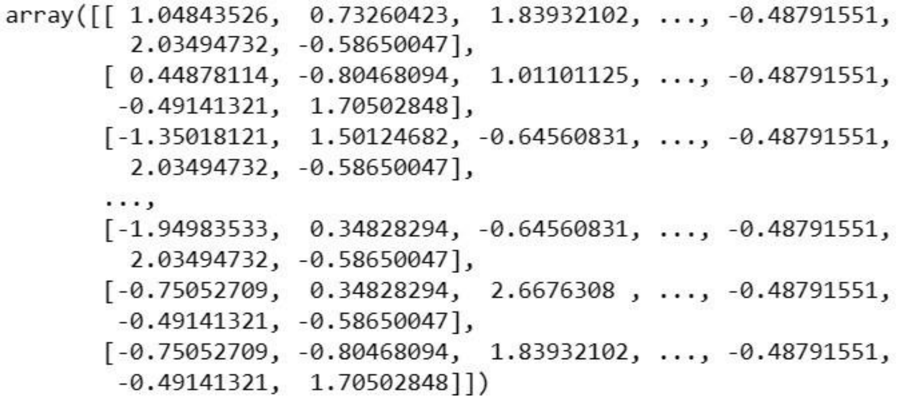
Table 1.7

**Modeling :**

- Chosen a variety of regression models including Linear Regression, Lasso, Ridge, K-Neighbors Regressor, Decision Tree, Random Forest Regressor, XG Booster, Cat Boosting Regressor, and Ada Boost Regressor.
- Train each selected model using the provided dataset.
- Evaluate the performance of each model on both the training and testing datasets.
- Predict the target variable "product_wg_ton" for both the training and testing datasets using each trained model.
- Assess the performance metrics, including Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R2 Score, for each model.
- Compare the performance of the different regression models based on their R2 Score and RMSE values.

**Splitting the Data:**

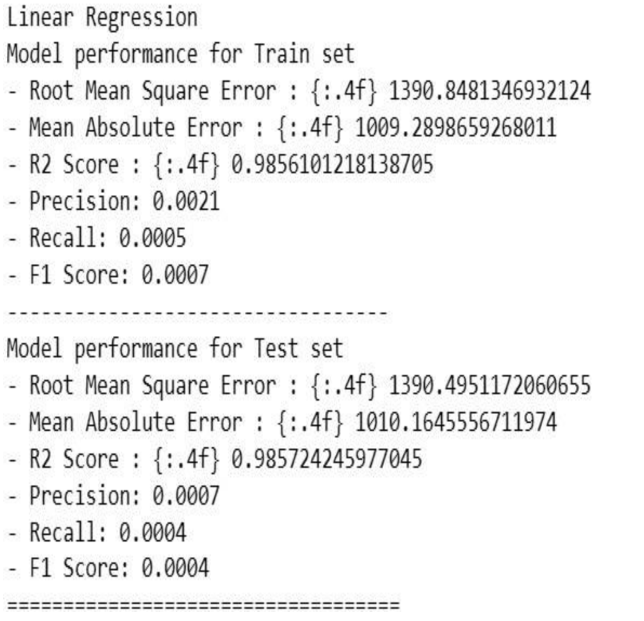
- Split the Data into 70:30 for train : test, respectively.
- The dataset is divided into training and testing sets, with 70% of the data allocated for training and 30% for testing. These split datasets are then employed to assess the performance of different models.



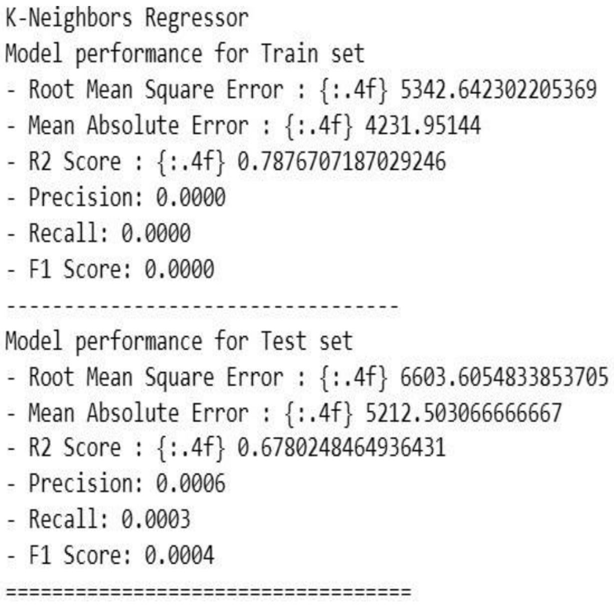
```
array([[ 1.04843526,  0.73260423,  1.83932102, ..., -0.48791551,
         2.03494732, -0.58650047],
       [ 0.44878114, -0.80468094,  1.01101125, ..., -0.48791551,
        -0.49141321,  1.70502848],
       [-1.35018121,  1.50124682, -0.64560831, ..., -0.48791551,
         2.03494732, -0.58650047],
       ...,
       [-1.94983533,  0.34828294, -0.64560831, ..., -0.48791551,
         2.03494732, -0.58650047],
       [-0.75052709,  0.34828294,  2.6676308 , ..., -0.48791551,
        -0.49141321, -0.58650047],
       [-0.75052709, -0.80468094,  1.83932102, ..., -0.48791551,
        -0.49141321,  1.70502848]])
```

Table 1.8

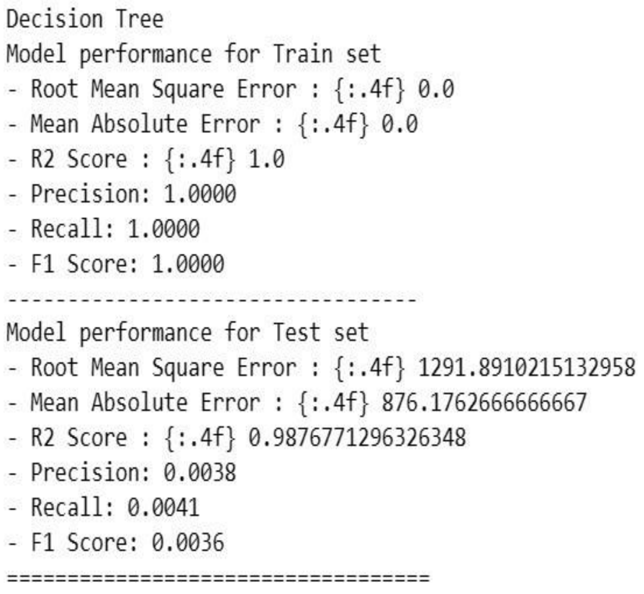
**Model Results :**



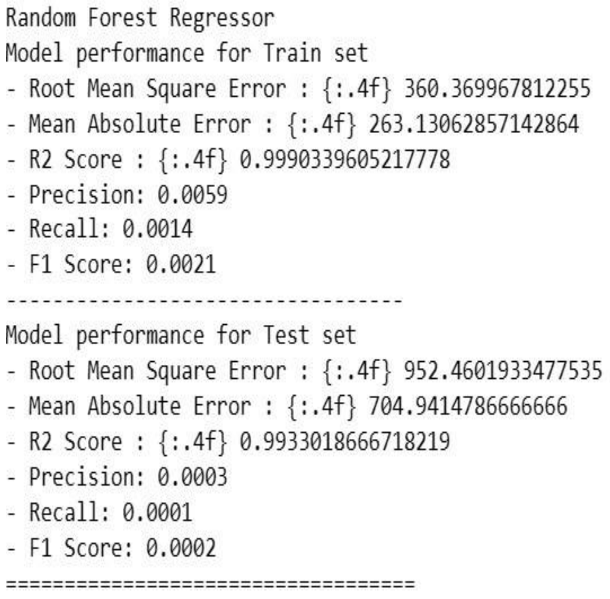
```
Linear Regression
Model performance for Train set
- Root Mean Square Error : {:.4f} 1390.8481346932124
- Mean Absolute Error : {:.4f} 1009.2898659268011
- R2 Score : {:.4f} 0.9856101218138705
- Precision: 0.0021
- Recall: 0.0005
- F1 Score: 0.0007
----------------------------------
Model performance for Test set
- Root Mean Square Error : {:.4f} 1390.4951172060655
- Mean Absolute Error : {:.4f} 1010.1645556711974
- R2 Score : {:.4f} 0.985724245977045
- Precision: 0.0007
- Recall: 0.0004
- F1 Score: 0.0004
===================================
```



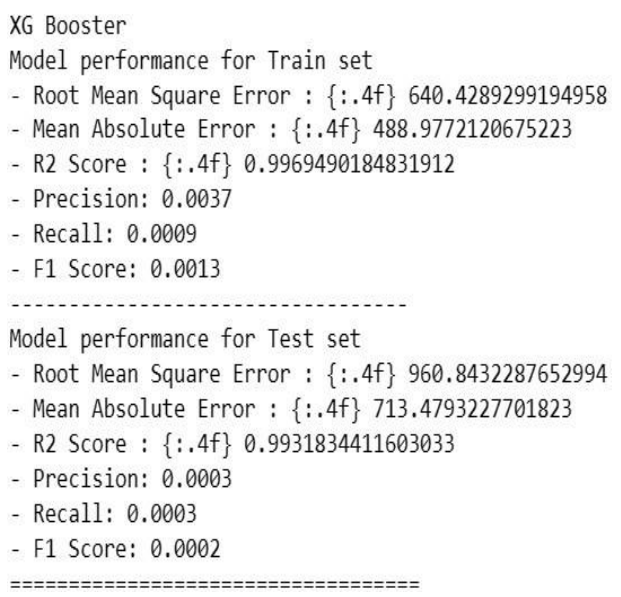
```
K-Neighbors Regressor
Model performance for Train set
- Root Mean Square Error : {:.4f} 5342.642302205369
- Mean Absolute Error : {:.4f} 4231.95144
- R2 Score : {:.4f} 0.7876707187029246
- Precision: 0.0000
- Recall: 0.0000
- F1 Score: 0.0000
----------------------------------
Model performance for Test set
- Root Mean Square Error : {:.4f} 6603.6054833853705
- Mean Absolute Error : {:.4f} 5212.503066666667
- R2 Score : {:.4f} 0.6780248464936431
- Precision: 0.0006
- Recall: 0.0003
- F1 Score: 0.0004
===================================
```



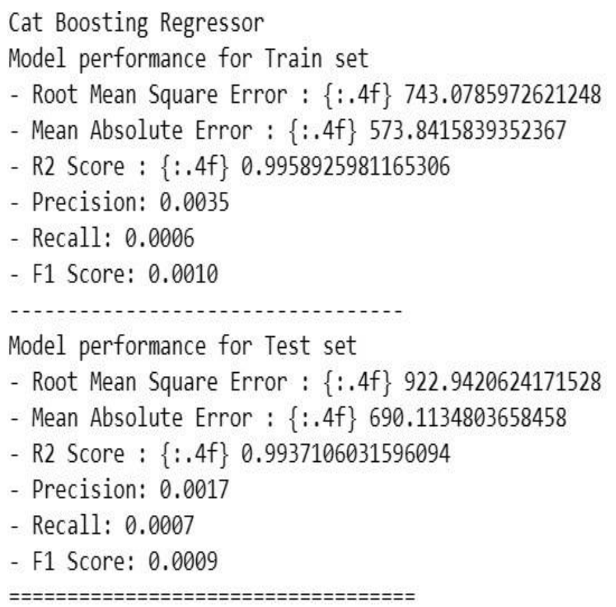
```
Decision Tree
Model performance for Train set
- Root Mean Square Error : {:.4f} 0.0
- Mean Absolute Error : {:.4f} 0.0
- R2 Score : {:.4f} 1.0
- Precision: 1.0000
- Recall: 1.0000
- F1 Score: 1.0000
----------------------------------
Model performance for Test set
- Root Mean Square Error : {:.4f} 1291.8910215132958
- Mean Absolute Error : {:.4f} 876.1762666666667
- R2 Score : {:.4f} 0.9876771296326348
- Precision: 0.0038
- Recall: 0.0041
- F1 Score: 0.0036
===================================
```



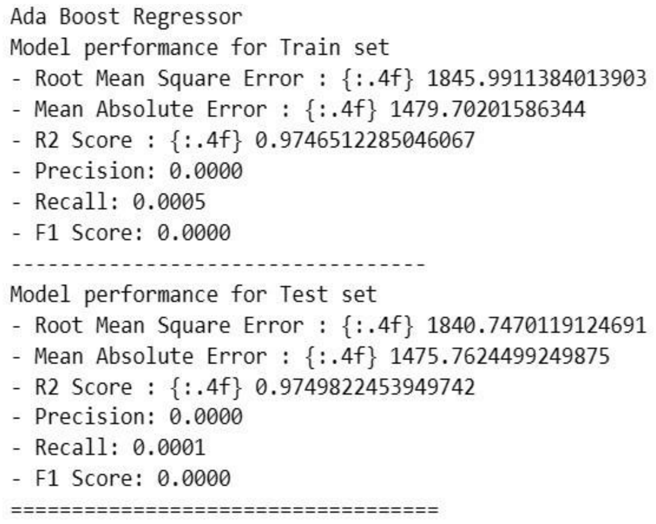
```
Random Forest Regressor
Model performance for Train set
- Root Mean Square Error : {:.4f} 360.369967812255
- Mean Absolute Error : {:.4f} 263.13062857142864
- R2 Score : {:.4f} 0.9990339605217778
- Precision: 0.0059
- Recall: 0.0014
- F1 Score: 0.0021
----------------------------------
Model performance for Test set
- Root Mean Square Error : {:.4f} 952.4601933477535
- Mean Absolute Error : {:.4f} 704.9414786666666
- R2 Score : {:.4f} 0.9933018666718219
- Precision: 0.0003
- Recall: 0.0001
- F1 Score: 0.0002
===================================
```



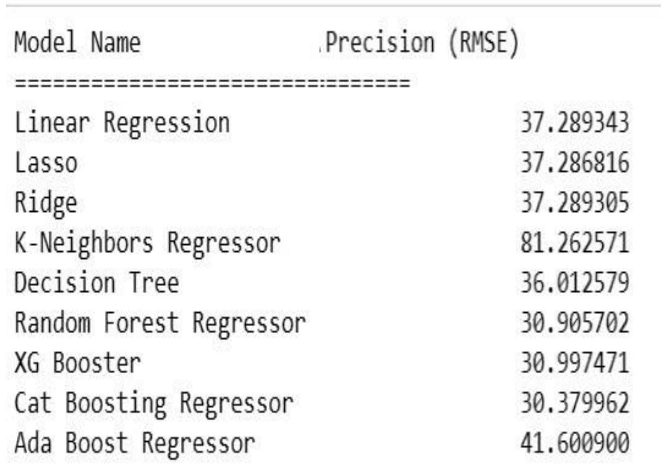
```
XG Booster
Model performance for Train set
- Root Mean Square Error : {:.4f} 640.4289299194958
- Mean Absolute Error : {:.4f} 488.9772120675223
- R2 Score : {:.4f} 0.9969490184831912
- Precision: 0.0037
- Recall: 0.0009
- F1 Score: 0.0013
----------------------------------
Model performance for Test set
- Root Mean Square Error : {:.4f} 960.8432287652994
- Mean Absolute Error : {:.4f} 713.4793227701823
- R2 Score : {:.4f} 0.9931834411603033
- Precision: 0.0003
- Recall: 0.0003
- F1 Score: 0.0002
===================================
```



```
Cat Boosting Regressor
Model performance for Train set
- Root Mean Square Error : {:.4f} 743.0785972621248
- Mean Absolute Error : {:.4f} 573.8415839352367
- R2 Score : {:.4f} 0.9958925981165306
- Precision: 0.0035
- Recall: 0.0006
- F1 Score: 0.0010
----------------------------------
Model performance for Test set
- Root Mean Square Error : {:.4f} 922.9420624171528
- Mean Absolute Error : {:.4f} 690.1134803658458
- R2 Score : {:.4f} 0.9937106031596094
- Precision: 0.0017
- Recall: 0.0007
- F1 Score: 0.0009
===================================
```



```
Ada Boost Regressor
Model performance for Train set
- Root Mean Square Error : {:.4f} 1845.9911384013903
- Mean Absolute Error : {:.4f} 1479.70201586344
- R2 Score : {:.4f} 0.9746512285046067
- Precision: 0.0000
- Recall: 0.0005
- F1 Score: 0.0000
----------------------------------
Model performance for Test set
- Root Mean Square Error : {:.4f} 1840.7470119124691
- Mean Absolute Error : {:.4f} 1475.7624499249875
- R2 Score : {:.4f} 0.9749822453949742
- Precision: 0.0000
- Recall: 0.0001
- F1 Score: 0.0000
===================================
```

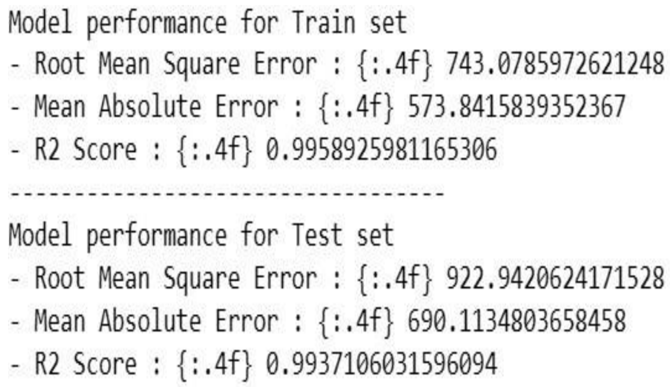


```
Model Name                Precision (RMSE)
==================================
Linear Regression                37.289343
Lasso                            37.286816
Ridge                            37.289305
K-Neighbors Regressor            81.262571
Decision Tree                    36.012579
Random Forest Regressor          30.905702
XG Booster                       30.997471
Cat Boosting Regressor           30.379962
Ada Boost Regressor              41.600900
```

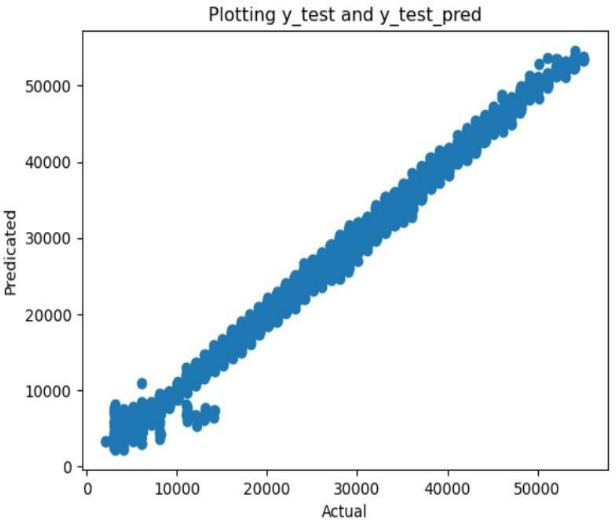
**Tuning : (Grid Search)**

Comparing the results of best performing model before and after tuning :

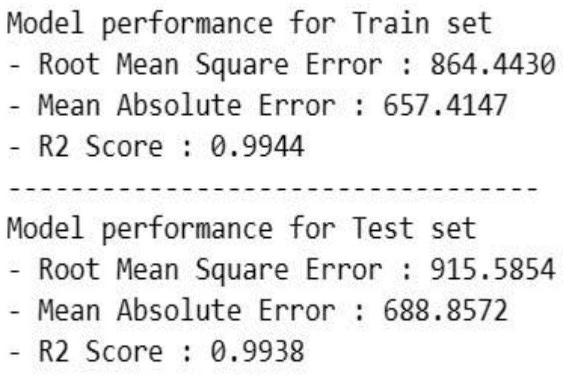
Before Tuning



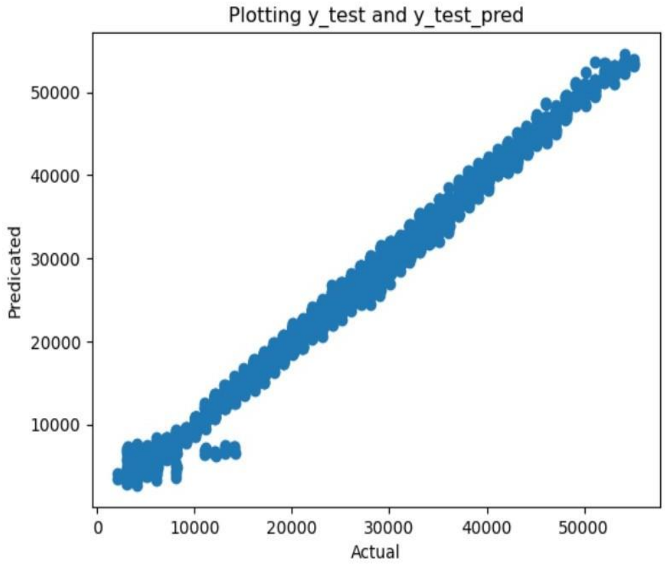
```
Model performance for Train set
- Root Mean Square Error : {:.4f} 743.0785972621248
- Mean Absolute Error : {:.4f} 573.8415839352367
- R2 Score : {:.4f} 0.9958925981165306
----------------------------------
Model performance for Test set
- Root Mean Square Error : {:.4f} 922.9420624171528
- Mean Absolute Error : {:.4f} 690.1134803658458
- R2 Score : {:.4f} 0.9937106031596094
```



After Tuning :



```
Model performance for Train set
- Root Mean Square Error : 864.4430
- Mean Absolute Error : 657.4147
- R2 Score : 0.9944
-----------------------------------
Model performance for Test set
- Root Mean Square Error : 915.5854
- Mean Absolute Error : 688.8572
- R2 Score : 0.9938
```



- Grid Search Cross-Validation tuning method used.

- Grid Search Cross-Validation helps in finding the optimal combination of hyper-parameters for the Cat Boost Regressor model, leading to improved performance compared to using default hyper-parameter values.

- Tuning have increased the performance of the model slightly.

## 1.6) Model validation.

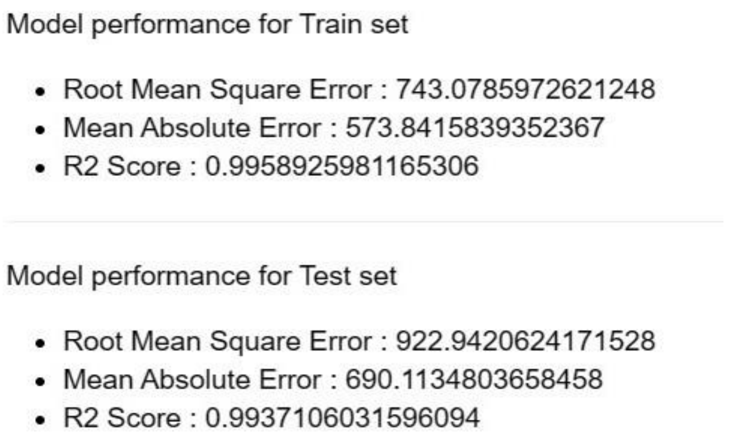
* Among the models, Cat Boost stands out as the top performer.

√ The Cat Boost has the highest R2 score

√ The MAE (Mean absolute error) is the lowest

√ And the model has the lowest RMSE.

Therefore, it's recommended to prioritize the implementation of Cat Boosting due to its superior predictive capabilities.

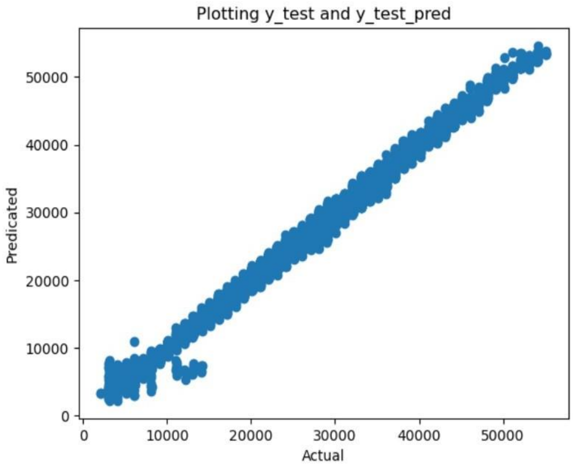


Model performance for Train set

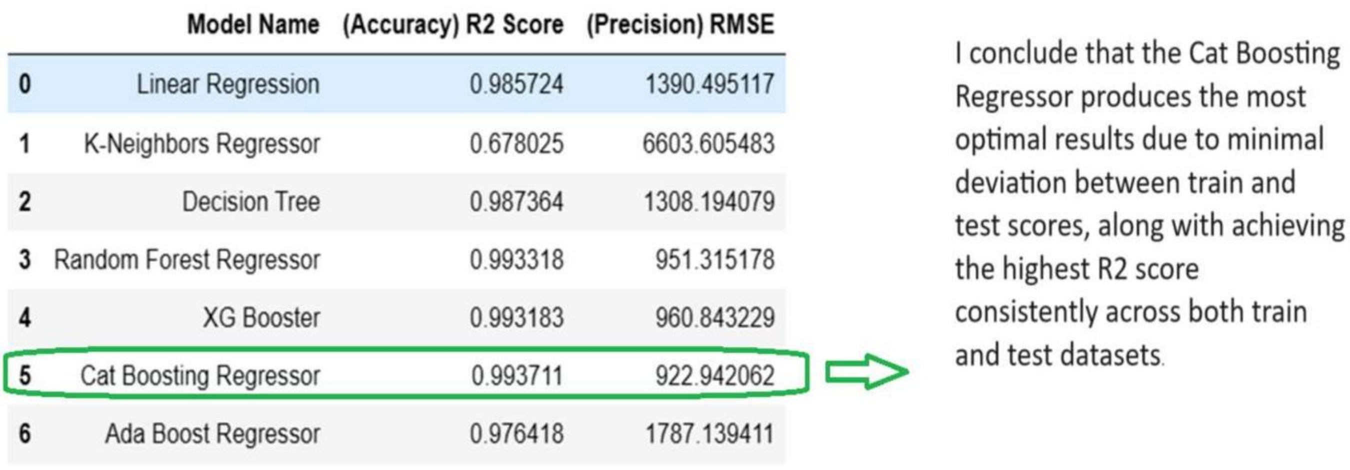
- Root Mean Square Error : 743.0785972621248
- Mean Absolute Error : 573.8415839352367
- R2 Score : 0.9958925981165306

Model performance for Test set

- Root Mean Square Error : 922.9420624171528
- Mean Absolute Error : 690.1134803658458
- R2 Score : 0.9937106031596094



* While Cat Boost seems best for now, it's essential to keep testing and refining models as the supply chain data changes or as we learn more about what works best for our specific situation.



| | Model Name | (Accuracy) R2 Score | (Precision) RMSE |
|---|---|---|---|
| 0 | Linear Regression | 0.985724 | 1390.495117 |
| 1 | K-Neighbors Regressor | 0.678025 | 6603.605483 |
| 2 | Decision Tree | 0.987364 | 1308.194079 |
| 3 | Random Forest Regressor | 0.993318 | 951.315178 |
| 4 | XG Booster | 0.993183 | 960.843229 |
| 5 | Cat Boosting Regressor | 0.993711 | 922.942062 |
| 6 | Ada Boost Regressor | 0.976418 | 1787.139411 |

I conclude that the Cat Boosting Regressor produces the most optimal results due to minimal deviation between train and test scores, along with achieving the highest R2 score consistently across both train and test datasets.

Table 1.10

* Implement Cat Boosting model for predicting demand variations and optimizing inventory levels across warehouses.

* Also establish a process for continuous monitoring and refinement of the Cat Boosting model. Regularly update the model with new data and refine algorithms to adapt to changing market dynamics and evolving business requirements.

## 1.7) Final interpretation / recommendation.

- From the data we can see, when the product shipment is more the storage issue and the warehouse shutdown is increasing and vice versa. Need to reduce the shipment in Zone 6, Zone5 and Zone 4; especially in North and west region. Also need to promote the on time delivery to avoid the storage issue in these regions. Delivery time plays important role.

- Similarly warehouse need to be instructed to stop over stocking of goods and to store required amount of goods in warehouse in North and West regions, this helps in company's growth.

- By reducing the number of refills, we can reduce the possibility of transportation issue. Need to also we can see the transportation issue are the highest in Zone 6, Zone 5 and Zone 4. Need to reduce the number of refills and increase the transportation capacity.

- Shipments are at their lowest in the Eastern region, which is also reflected in the sales figures. To improve sales, increasing advertising efforts in this region is recommended, especially considering the relatively lower number of competitors.

- Implement Cat Boosting model for predicting demand variations, optimizing inventory levels across warehouses. To improve distribution planning by integrating predictions from the Cat Boosting Regressor model into logistics and transportation management systems.

- Establish a process for continuous monitoring and refinement of the Cat Boosting Regressor model. Regularly update the model with new data and refine algorithms to adapt to changing market dynamics and evolving business requirements.

By implementing these recommendations based on the insights derived from the Cat Boosting Regressor model, businesses can achieve significant improvements in supply chain efficiency, inventory management, customer satisfaction, and overall profitability.