2024

Capstone Project

SUPPLY CHAIN ANALYTICS

RANJITH RAMESH

Table of Contents

Table of Figures

Table of Tables

Introduction

Business Problem:

A FMCG company has entered the instant noodles business two years back. Their higher management has noticed a mismatch 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 cases, it results in inventory cost loss to the company. Hence, the higher management wants to optimize the supply quantity in each warehouse in the entire country.

Goal & Objective:

The objective of this exercise is to build a model using historical data to determine the optimum weight of the product to be shipped to each warehouse. Additionally, the company aims to analyse the demand pattern in different pockets of the country so that management can drive advertisement campaigns particularly in those areas.

Need of the Study/Project:

1. Inventory Cost Optimization: By optimizing the supply quantity in each warehouse, the company can minimize inventory cost losses resulting from mismatched demand and supply.
2. Improved Efficiency: With an optimized supply chain, the company can improve operational efficiency by ensuring that each warehouse receives the appropriate quantity of product based on demand.
3. Strategic Decision Making: Analysing demand patterns across different regions allows management to make informed decisions regarding advertisement campaigns, focusing efforts on areas with higher potential demand.

Competitive Advantage: By addressing supply chain inefficiencies and strategically targeting advertisement campaigns, the company can gain a competitive advantage in the market.

Exploratory Data Analysis

Uni-Variate Analysis:

Government rating of warehouse:

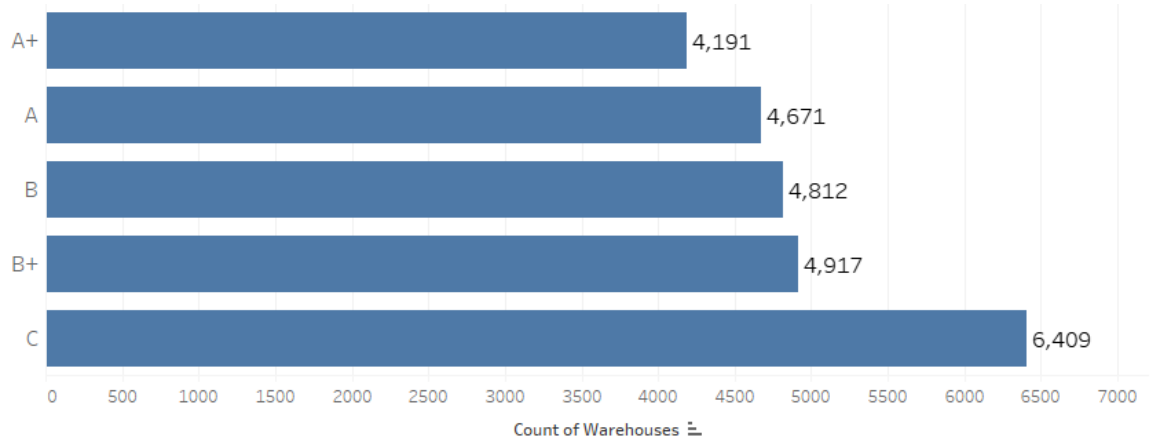


Figure 1: Bar Graph of Govt Rating of Warehouses

Inference:

- This feature is considered as categorical feature, which represents the government rating provided to the warehouse, it can take any one value from A+, A, B+, B and C.
- It can be depicted from the bar graph that, most warehouses (6409) are rated C, which is the lowest rating compared to other types of rating.
- However, 4191 warehouses have been rated A+, which is the highest form of rating.

Storage Issues reported in last 3 months:

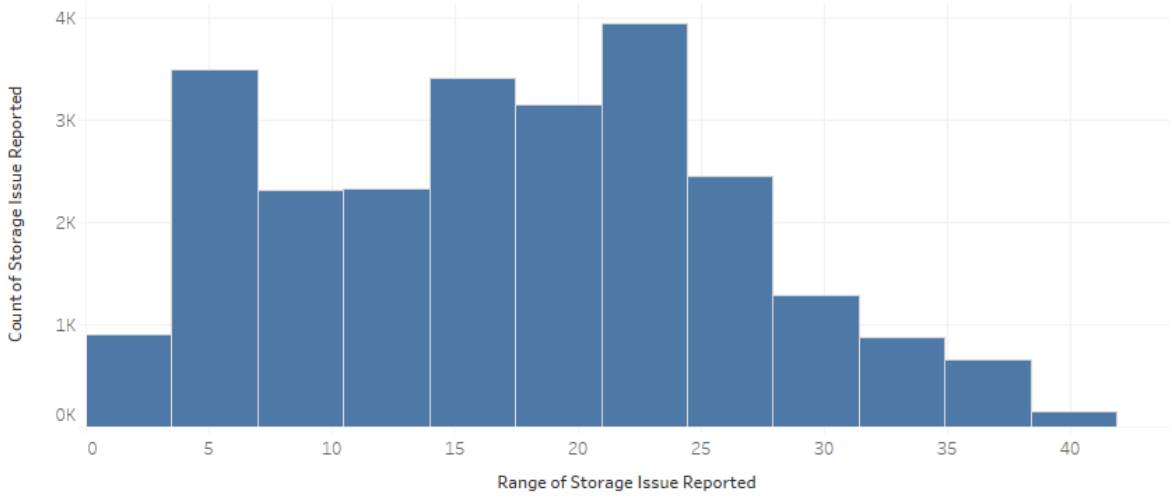


Figure 2: Histogram of Storage Issues Reported

Inference:

- This feature represents the number of storage issues reported in the last 3 months, it's a continuous numerical feature ranging from 0 to 45 approximately.

Competitor in market

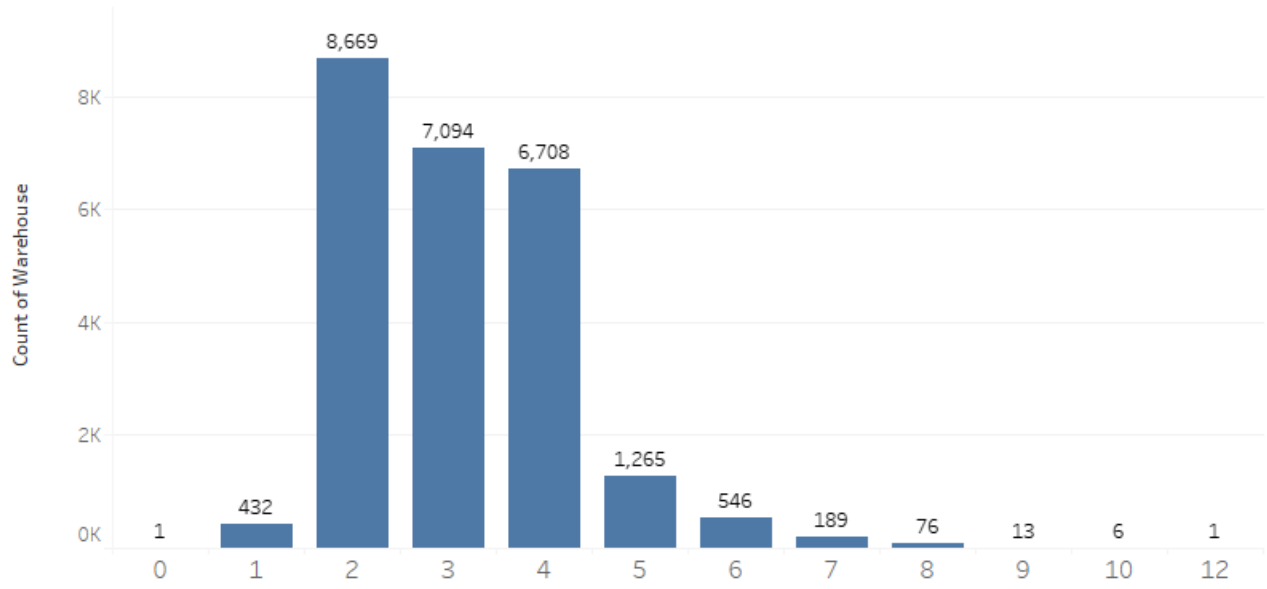


Figure 3: Bar Graph of # of Competitor

Inference:

- This feature is considered categorical, which could take any values from the range 0 to 12, each representing the number of instant noodles competitor in the market.
- It can be depicted from that bar graph; majority of warehouses has 2 to 4 competitors,
- 1265 warehouses have around 5 competitors, 546 warehouses have around 6 competitors

Transport Issues reported in last 1 year:

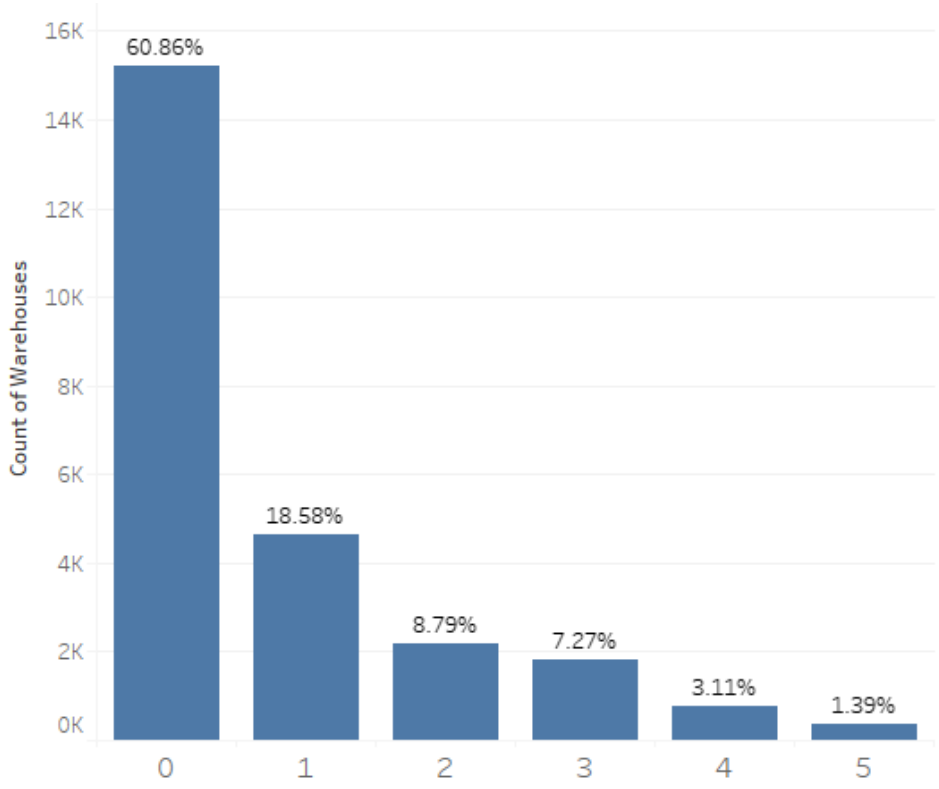


Figure 4: Bar Graph of Transport Issues

Inference:

- This feature is categorical, with values ranging from 0 to 5, representing the number of times the warehouse has experienced transportation issues in the last year.
- The bar graph illustrates the count of warehouses within each category.
    - It can be inferred that the majority of warehouses (60.86%) have not experienced any transportation issues in the last year.
    - Warehouses that experienced 1 transportation issue account for 18.58%.
    - Those with 2 and 3 transportation issues represent 8.79% and 7.27%, respectively.
    - The warehouses experiencing the highest number of transportation issues, 5, constitute 1.39% of the entire dataset.

Retail Shop Number:

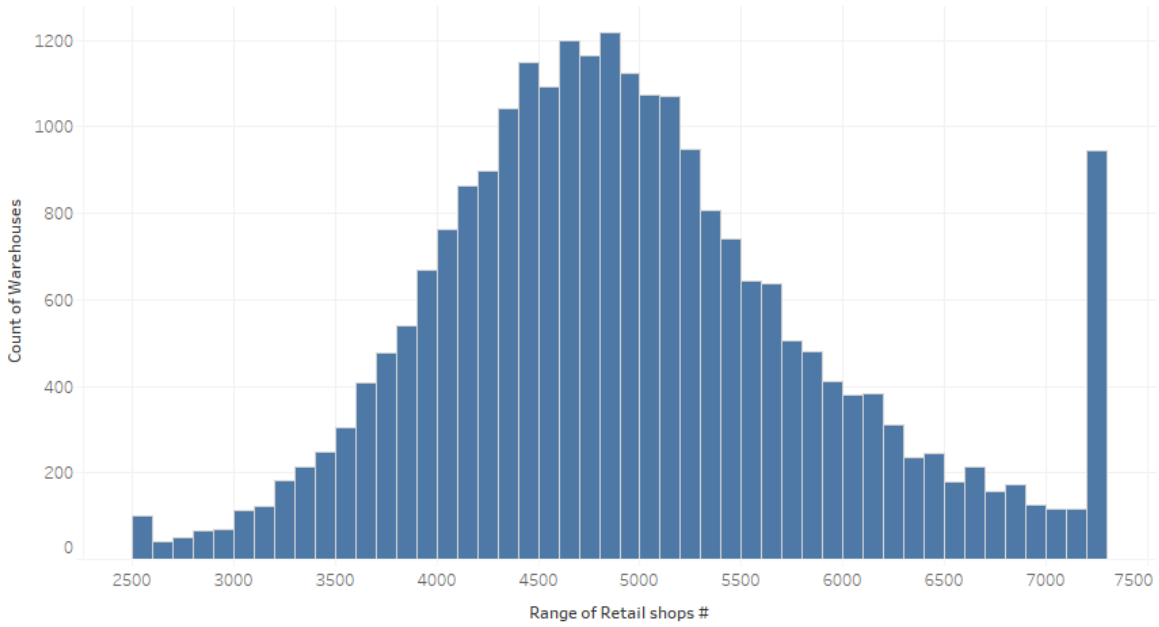


Figure 5: Histogram of Retail Shop Number

Inference:

- This feature represents the number of retail shop that sells the product under the warehouse area, it is considered as continuous numerical feature.
- The distribution of the number is close to normal distribution.

Reason for the spike in values at the end is due to capping of outliers at lower and upper bound values calculated by IQR method.

Electric Supply:

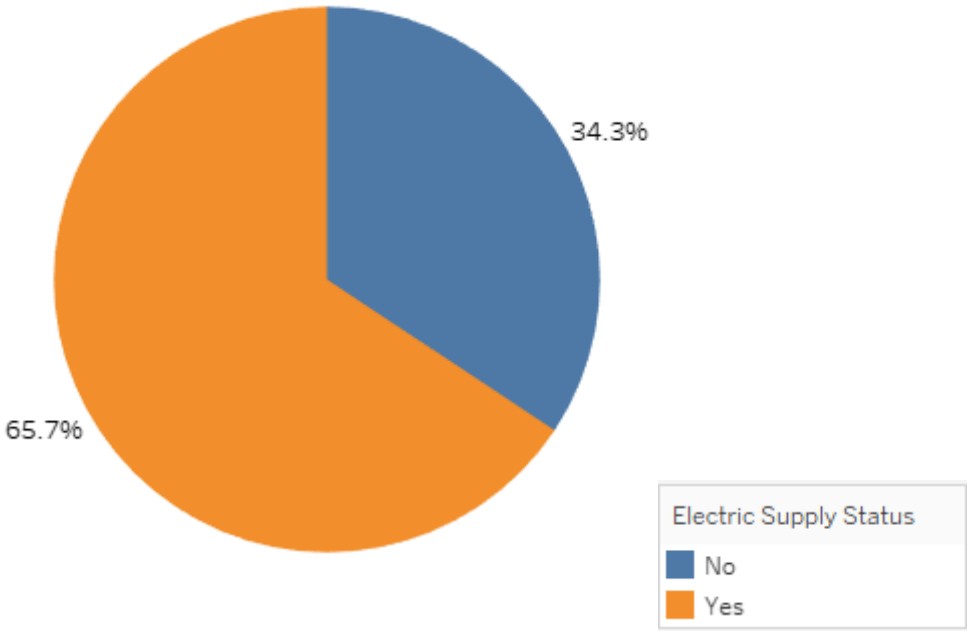


Figure 6: Pie Chart of Electric Supply

Inference:

This features is considered as categroical feature, which represents the status of whether the the warehouse is equiped with electricity back up like generator, its denoted by the choices, “Yes” and “No”.

Product weight in tons shipped in last 3 months:

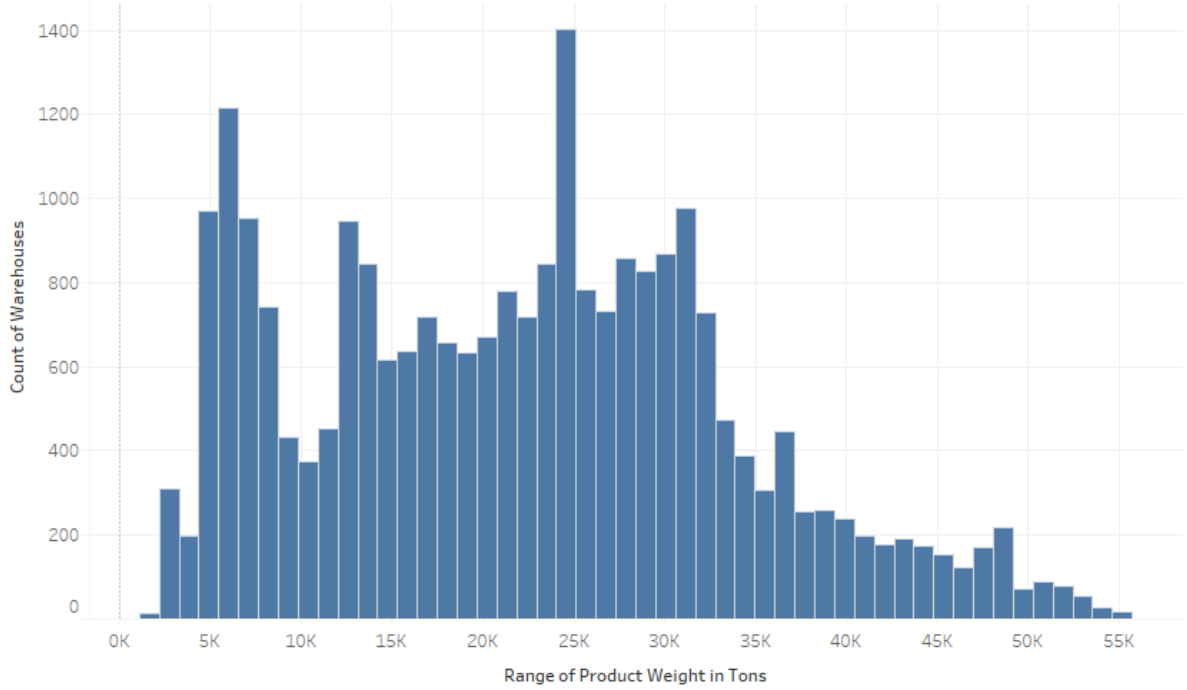


Figure 7: Histogram of Product Weight shipped in last 3M (Tons)

Inference:

- This feature is considered as continuous numerical feature, it's the dependent feature, which represents the weights in tone shipped to the warehouse.
- Its distribution is almost close to that of a normal distribution.

Bi-Variate Analysis:

Count plot of Zones vs Capacity:

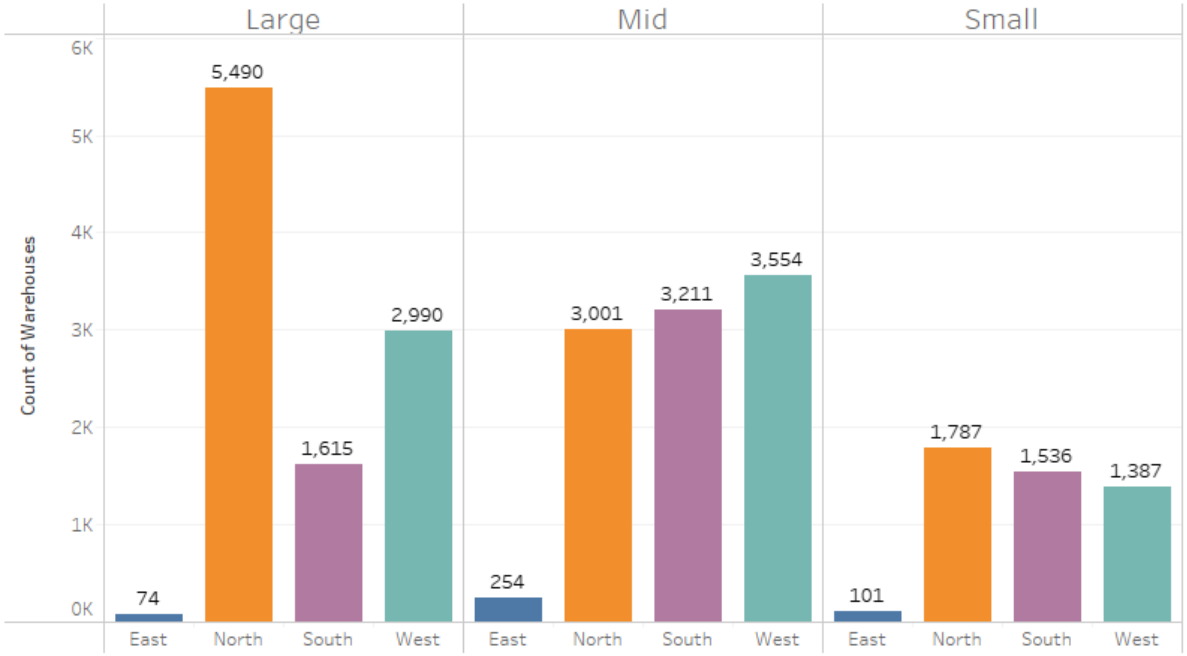


Figure 8: Warehouse (Zone Vs Capacity)

Inference:

- We depict a bar graph representing the count of warehouses between different zones and their capacity to understand the distribution between different zones.
- It can be inferred; Majority of large warehouses (5490) are located in the north region
- When it comes to mid and small sized warehouses, there is an almost similar level of splits between zones (except east)
- The east accounts for the least number of warehouses in all 3-size category.

Competitors in different zones:

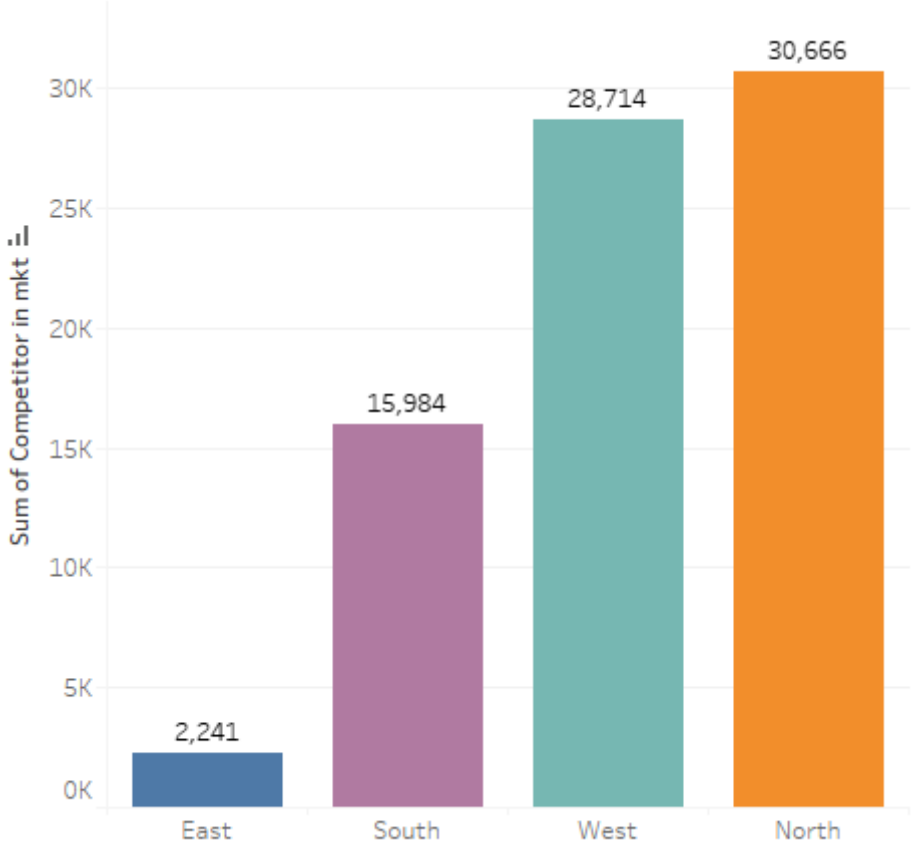


Figure 9: Zone vs Competitors

Inference:

- It can be denoted that North Zone leads in having the number of competitors with 30,666.
- Followed up by West with 28,714
- The south and East zone have 15,984 and 2,241 competitors respectively.

Warehouse electricity backup vs temperature regulator machine

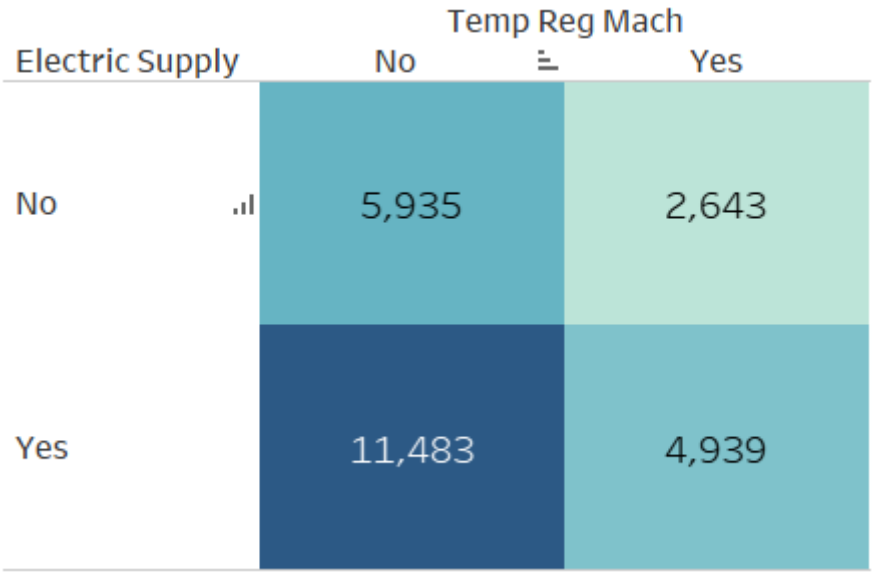


Figure 10 Crosstab of Electric Supply Vs Temp Reg Machine

Inference:

- It can be inferred from the crosstab that majority of warehouses (11,483) have electricity supply backup but no temperature regulator machine and minority of warehouses have temperature regulator machine but no electricity supply backup.
- There are 5935 warehouses which doesn't have both.
- There are 4939 warehouse which does have both.

Warehouse Flood Impact vs Flood Proof:

Flood Impacted	Flood Proof: No	Flood Proof: Yes
No	21,495	1,051
Yes	2,139	315

Figure 11: Crosstab of Flood Impacted vs Proof

Inference:

- The cross tab depicts the split between warehouses which were impacted by flood and considered as proofed for flood.
- 21,495, Majority of the warehouses were not impacted by flood or flood proofed.
- 315 warehouses were flood proofed and were hit with the flood.
- 2139 warehouses were impacted by flood and were not flood proofed
- 1051 warehouses were flood proofed but didn’t get hit by the flood.

Pair plot of all the numerical features:

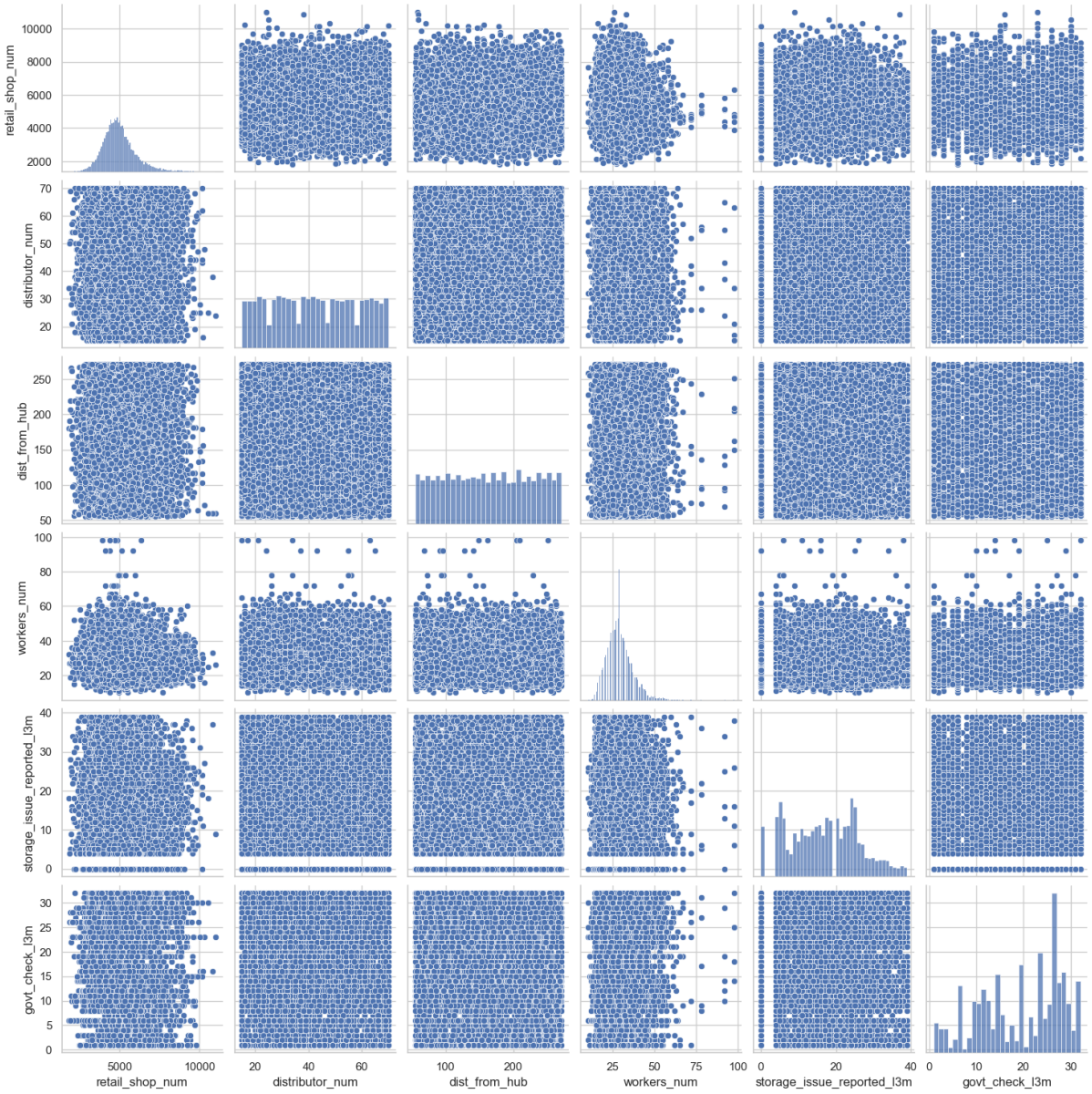


Figure 12: Pair plot of numerical features

Inference:

- There isn't any correlation of any kind visible in the numerical features.
- They seem to possess an ordinary & random distribution.

Data Cleaning and Pre-processing

Approach used for identifying and treating missing values and outlier treatment

Missing Value Treatment:

There are two columns with missing values, “workers_num” – continuous numerical variable and “approved_wh_govt_certificate” – categorical variable.

Before treating missing values:

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	990
storage_issue_reported_l3m	0
temp_reg_mach	0
approved_wh_govt_certificate	908
wh_breakdown_l3m	0
govt_check_l3m	0
product_wg_ton	0

After treating missing values:

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
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

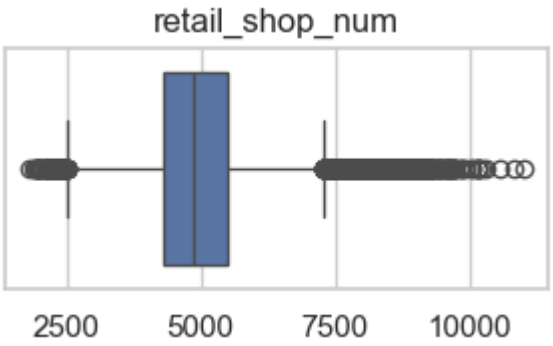
“workers_num” – continuous numerical variable, the missing values are treated by imputing the mean of the feature, since it’s data distribution is close to that of normal distribution.

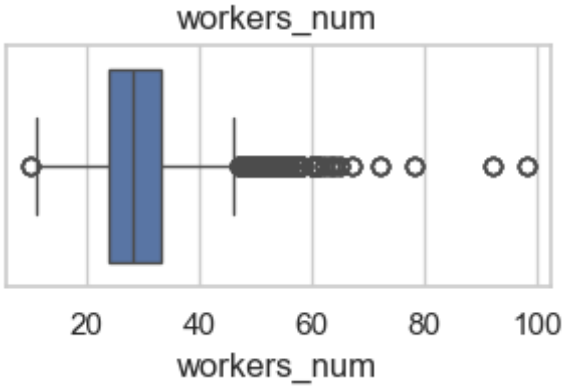
“approved_wh_govt_certificate” – categorical variable, the missing values are treated by imputing the mode of the feature, since its categorical feature.

Outlier treatment:

By plotting the boxplot of the numerical feature, we identify ouliters being present in only two of the features, which are “retail_shop_num” and “workers_num”

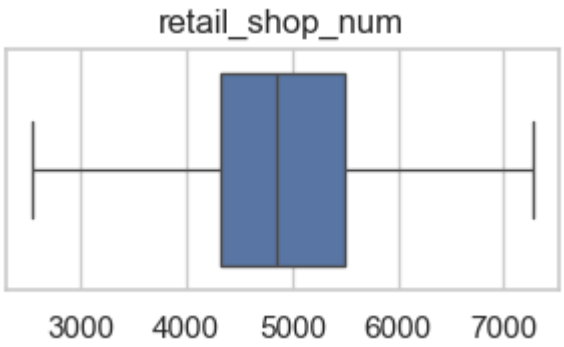
Before outlier treatment:

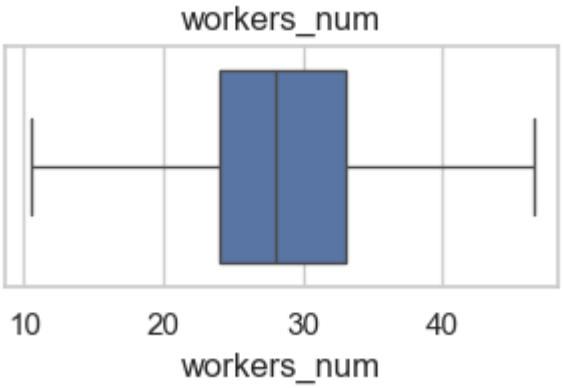




We decide to perform capping by the lower and upper bound values calculated by the inter-quartile range method for both the features respectively instead of dropping the outlier to prevent information loss.

After outlier treatment:





Need for variable transformation

The categorical features are encoded using a label encoder and numerical features are standardized to fit in the kmeans clustering model. The preview of the categorical features is provided in the juypter notebook.

Variables removed or added and why

The features "Ware_house_ID" and "WH_Manager_ID" are removed as they are unique identifiers used to ID the warehouse and manager, which doesn’t serve any purpose for the model building.

Later we have chosen to eliminate "wh_est_year" since it only provides details about when the warehouse started and a significant percentage of data for that feature is missing.

Model Building

Classification approach to solve the business problem:

Given the absence of binary features indicating understocking or overstocking status, we devised an alternative approach. Initially, we clustered the dataset to identify warehouses exhibiting similar characteristics. Subsequently, we assigned labels to these clusters. Our assumption posits that the mean of these clusters represents the average demand. With this premise in mind, we proceeded to construct classification models. The primary objective of these models is to predict the class of a warehouse based on its features. Ultimately, this facilitates recommendations regarding appropriate shipping capacities for the respective warehouses.

Different Classification Models

Since we have more than two clusters, we will be implementing multiclass classification models for this dataset. We have already conducted preprocessing steps, including treatment of nulls and outliers, encoding, and scaling. Our intention is to build a wide variety of models and compare them based on several metrics to determine the best model for this dataset.

The dataset was divided into training and testing sets using a 70:30 split ratio. From clustering we have produced 3 clusters labelled 0,1 and 2.

Varity of models include:

Base Models:

- Decision Tree Classifier
- Extra Tree Classifier
- K-Neighbors Classifier

Ensemble Models:

- Bagging Decision Tree Classifier
- Ensemble Extra Tree Classifier
- Ensemble Random Forest Classifier
- Ada Boost Classifier
- Gradient Boost Classifier

Why do we use Gradient Boosting Classifier?

Gradient Boosting Classifier is an ensemble learning method that sequentially trains weak learners to correct errors made by previous models. It fits new learners to residuals of previous predictions, optimizes their parameters using gradient descent, and combines their predictions to make the final one. Regularization techniques like shrinkage and tree depth constraints are applied to prevent overfitting. Training continues until a specified number of learners are trained or a stopping criterion is met. Overall, Gradient Boosting builds a powerful model by iteratively improving predictions and achieving high accuracy on diverse tasks.

Confusion Matrix:

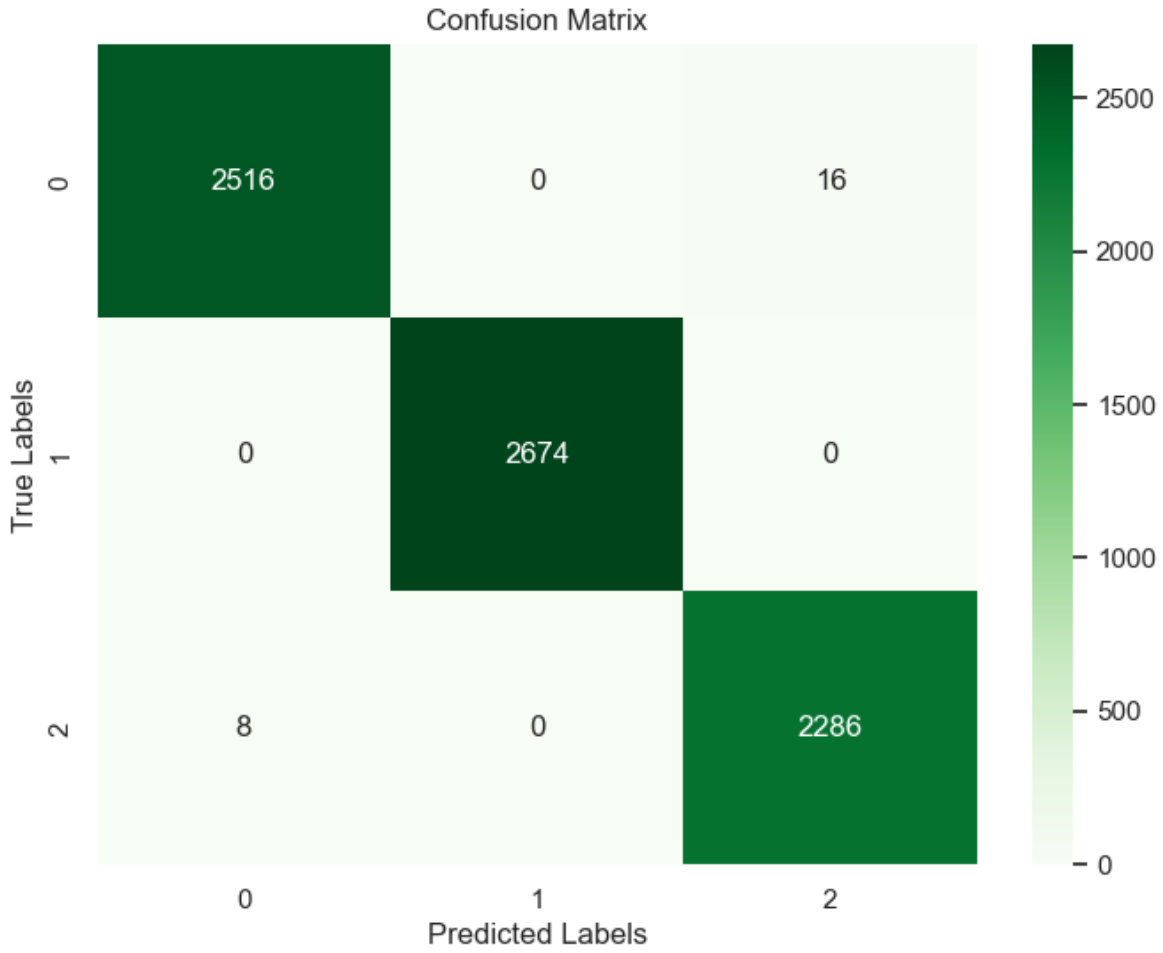


Figure 13: Confusion Matrix of Gradient Boosting Classifier

Inference from Confusion Matrix:

1. **Class 0 (Label 0)**:
    - 2516 instances are correctly classified as Class 0.
    - 16 instances of Class 0 are misclassified as Class 2.
2. **Class 1 (Label 1)**:

- All 2674 instances of Class 1 are correctly classified.

3. **Class 2 (Label 2)**:
    - 2286 instances are correctly classified as Class 2.
    - 8 instances of Class 2 are misclassified as Class 0.

The confusion matrix reveals that the Gradient Boosting Classifier performs well, achieving high accuracy with minimal misclassifications. Most instances are correctly classified, with only a small number of misclassifications observed between Class 0 and Class 2. Further analysis may be needed to understand the patterns of misclassification and potential improvements.

Classification Report:

	precision	recall	f1-score	support
0	1.00	0.99	1.00	2532
1	1.00	1.00	1.00	2674
2	0.99	1.00	0.99	2294
accuracy			1.00	7500
macro avg	1.00	1.00	1.00	7500
weighted avg	1.00	1.00	1.00	7500

Table 1: Classification Report of Gradient Boost Classifier

Inference from Classification Report:

The classification report provides insights into the Gradient Boosting Classifier's performance across various metrics:

1. **Precision**:
    - Class 0: The precision for Class 0 is 1.00, indicating that all instances predicted as Class 0 are indeed Class 0.
    - Class 1: The precision for Class 1 is 1.00, signifying perfect precision where all instances predicted as Class 1 are correct.
    - Class 2: The precision for Class 2 is 0.99, indicating that 99% of instances predicted as Class 2 are indeed Class 2.
2. **Recall**:
    - Class 0: The recall for Class 0 is 0.99, implying that 99% of actual Class 0 instances are correctly classified.
    - Class 1: The recall for Class 1 is 1.00, suggesting that all actual Class 1 instances are correctly classified.
    - Class 2: The recall for Class 2 is 1.00, indicating that all actual Class 2 instances are correctly classified.

3. **F1-score**:
    - The F1-score, which is the harmonic mean of precision and recall, is high for all classes, indicating a good balance between precision and recall.
4. **Accuracy**:
    - The overall accuracy of the model is 1.00 (or 100%), suggesting that all of the model's predictions are correct across all classes.
5. **Macro Average**:
    - Both macro average precision, recall, and F1-score are 1.00, calculated by averaging the scores for each class equally, suggesting consistent and perfect performance across classes.
6. **Weighted Average**:
    - The weighted average precision, recall, and F1-score are also 1.00, taking into account the support for each class, further confirming the model's perfect performance across the dataset.

Overall, the Gradient Boosting Classifier demonstrates exceptional performance with high precision, recall, and F1-scores for all classes, indicating its accuracy and reliability for classification tasks.

Efforts to improve the classfication model performance:

Fixing the ***learning rate (0.1)*** to yield maximum accuracy, which is a hyperparameter that controls the contribution of each tree to the final prediction. It scales the contribution of each tree by the value of the learning rate.

We tune the **max depth to 8**, which is a parameter controls the maximum depth of each individual tree in the ensemble. Depth refers to the length of the longest path from the root node to a leaf node in the tree. Its significant in controlling model complexity, preventing overfitting, computational efficiency and interpretability.

Setting the ***n_estimators to 100***, which is a parameter that controls the number of weak learners (individual decision trees) to be used in the ensemble. Each weak learner is added sequentially to the ensemble, and they collectively form a strong learner through iterative training. It overall helps in improving the balancing in bias and variance, regularization, computational complexity and model performance.

Classification Model Validation:

The dataset was divided into training and testing sets using a 70:30 split ratio. From clustering we have produced 3 clusters labelled 0,1 and 2.

Our intention is to build a wide variety of models and compare them based on several metrics such as precision, recall and accuracy to determine the best model for this dataset.

Models	Class	Precision	Recall	Accuracy
Decision Tree Classifier	**0**	0.99	0.99	99.4
	1	1.00	1.00	
	2	0.99	0.99	
Extra Tree Classifier	**0**	0.90	0.92	93
	1	0.95	0.95	
	2	0.93	0.91	
K-Nearest Neighbours Classifier	**0**	0.69	0.78	74
	1	0.74	0.65	
	2	0.83	0.84	
Bagging Decision Tree	**0**	0.99	0.99	99.5
	1	1.00	1.00	
	2	0.99	0.99	
Ensemble Extra Tree	**0**	0.98	0.99	99
	1	1.00	1.00	
	2	0.99	0.98	
Ensemble Random Forest	**0**	0.98	0.99	98.9
	1	1.00	1.00	
	2	0.98	0.98	
AdaBoost	**0**	0.95	1.00	97.9
	1	1.00	1.00	
	2	0.99	0.94	
Gradient Boosting	**0**	1.00	0.99	**99.6**
	1	1.00	1.00	
	2	0.99	1.00	

Table 2: Classification Model Comparison

So the best model is gradient boost, which is based on its accuracy, precision and recall, compared to all the other models.

Final Interpretation of Classification model:

Based on the above table, **Gradient Boost Classifier**, an ensemble model, has the ***best accuracy score***.

Feature importance of Gradient Boost Classifier:

Feature	Values
approved_wh_govt_certificate	**0.68**
storage_issue_reported_l3m	**0.18**
product_wg_ton	**0.12**
dist_from_hub	0.0050
workers_num	0.0001

Table 3: Feature Importance of Best Classifier Model (gb_classifier)

In a multiclass classifier decision tree, feature importance measures the contribution of each feature in the decision-making process across all classes. It assesses the significance of features in splitting the data and determining class labels. Higher feature importance suggests greater influence in class prediction, aiding in understanding the model's decision logic and identifying key factors driving classification outcomes.

Business Implication of the above classification model:

The classes formed from the initial clustering represent different groups of warehouses that are similar in nature, meaning they have similar "product_wg_ton" values, which represent the weights of shipments for the past 3 months.

We group these warehouses and determine their mean product weight, setting that number as the average demand for that type of warehouse. Thus, we obtain the following table:

Group	Average Product Weight in Tons
Class 0	28927.35
Class 1	25349.95
Class 2	10501.16

Table 4: Product Weight Table

Next, we proceed to construct multiple classification models that learn from the cluster-labelled data and predict the warehouse class based on its other features. Once we have built an accurate model, such as the Gradient Boost Classifier achieving 99.6% accuracy, whichever class it predicts, we designate the average product weight of that class as its average demand for the next 3-month period. We then advise the warehouse to allocate other resources accordingly to meet that demand.

Regression approach to solve the business problem:

In order to predict the optimum levels of product weight in tones, we use linear regression. Along with linear regression, we use other tree based regressors such as AdaBoost and Gradient Boost to obtain higher accuracy in prediction.

Why Choose Gradient Boost Regressor?

Comparing Gradient Boosting Regressor (GBR), AdaBoost Regressor, and Statsmodels Linear Regression involves understanding their strengths and weaknesses in different contexts. Here's a comparison to help you understand how GBR might be better than the others in certain scenarios:

1. **Flexibility in Modeling Nonlinear Relationships**:
    - Gradient Boosting Regressor: GBR can capture complex nonlinear relationships between predictors and the target variable by combining multiple weak learners (typically decision trees).
    - AdaBoost Regressor: AdaBoost can also capture nonlinear relationships, but it's generally not as flexible as GBR since it uses a weighted combination of weak learners.
    - Statsmodels Linear Regression: Linear regression assumes a linear relationship between predictors and the target variable, which may not adequately capture nonlinear patterns in the data.
2. **Handling of Heterogeneous Data**:
    - Gradient Boosting Regressor: GBR can handle a mix of numerical and categorical predictors without the need for encoding or preprocessing, making it more convenient for real-world datasets.
    - AdaBoost Regressor: Similar to GBR, AdaBoost can handle heterogeneous data, but it might require more preprocessing for categorical variables.
    - Statsmodels Linear Regression: Statsmodels linear regression requires encoding of categorical variables, which adds complexity and potential information loss.
3. **Robustness to Outliers**:
    - Gradient Boosting Regressor: GBR is generally robust to outliers due to its ensemble nature, which can reduce the impact of individual data points on the overall model.
    - AdaBoost Regressor: AdaBoost can be sensitive to outliers, especially when using decision trees as weak learners.
    - Statsmodels Linear Regression: Linear regression can be sensitive to outliers, as it tries to minimize the sum of squared errors, and outliers can have a significant impact on the regression coefficients.

4. **Model Interpretability**:
    - Gradient Boosting Regressor: GBR tends to be less interpretable compared to linear regression, as it involves combining multiple weak learners and might not provide easily interpretable coefficients.
    - AdaBoost Regressor: Similar to GBR, AdaBoost can be less interpretable due to its ensemble nature.
    - Statsmodels Linear Regression: Linear regression provides easily interpretable coefficients, making it straightforward to understand the relationship between predictors and the target variable.
5. **Performance on Large Datasets**:
    - Gradient Boosting Regressor: GBR can handle large datasets efficiently, especially with implementations like XGBoost and LightGBM, which are optimized for speed and scalability.
    - AdaBoost Regressor: AdaBoost might suffer from performance issues on large datasets, as it involves iterative training of multiple weak learners.
    - Statsmodels Linear Regression: Linear regression can handle large datasets efficiently, but it might not scale as well as gradient boosting methods for very large datasets.

In summary, Gradient Boosting Regressor can be advantageous over AdaBoost Regressor and Statsmodels Linear Regression in terms of flexibility, handling heterogeneous data, robustness to outliers, and performance on large datasets. However, the choice of algorithm depends on the specific characteristics of your data, the complexity of the relationships you're trying to model, and the importance of interpretability.

Efforts to improve the regressor model:

Fixing the ***learning rate (0.1)*** to yield maximum accuracy, which is a hyperparameter that controls the contribution of each tree to the final prediction. It scales the contribution of each tree by the value of the learning rate.

We tune the **max depth to 8**, which is a parameter controls the maximum depth of each individual tree in the ensemble. Depth refers to the length of the longest path from the root node to a leaf node in the tree. Its significant in controlling model complexity, preventing overfitting, computational efficiency and interpretability.

Setting the ***n_estimators to 100***, which is a parameter that controls the number of weak learners (individual decision trees) to be used in the ensemble. Each weak learner is added sequentially to the ensemble, and they collectively form a strong learner through iterative training. It overall helps in improving the balancing in bias and variance, regularization, computational complexity and model performance.

Regressor Model Validation

The three models constructed under the regression concept are linear regression using Stats models, AdaBoost Regressor, and Gradient Boosting Regressor. They are compared based on their ability to predict the test dataset. The comparison metrics include mean squared error (MSE), root mean squared error (RMSE), and R-squared value for each model. The model with the best metrics is selected as the final production model.

MSE, RMSE, and R2 are key metrics for evaluating regression models:

1. **MSE** measures the average squared difference between actual and predicted values, penalizing larger errors.
2. **RMSE** is the square root of MSE, providing an interpretable measure in the same units as the target variable.
3. **R2** indicates the proportion of variance in the target variable explained by the model, ranging from 0 to 1.

Lower MSE and RMSE values and higher R2 values signify better model performance.

Models	MSE	RMSE	R-Squared
Linear Regression (Statsmodel)	31,30,709	1,769	0.977
Ada Boost Regressor	10,41,846	1,021	0.992
Gradient Boost Regressor	***8,70,433***	***933***	***0.994***

Table 5: Regression Model Comparison

Final Interpretation of Regressor Model:

Based on the R-Squared and RMSE, Gradient Boost Regressor is selected. We further analyze the feature importance as it measures the contribution of each feature in the decision-making process across all classes. It assesses the significance of features in splitting the data and determining the optimum product weight in tons.

Higher feature importance suggests greater influence in weight prediction, aiding in understanding the model's decision logic and identifying key factors driving weight prediction outcomes.

Top 5 Features	Values
storage_issue_reported_l3m	**98.57%**
approved_wh_govt_certificate	**0.90%**
transport_issue_l1y	**0.12%**
temp_reg_mach	0.09%
retail_shop_num	0.06%
Rest of the feature (cumulative)	0.26%

Table 6: Feature Importance of Gradient Boost regressor

Recommendation:

Two approaches are available to address the problem. The client has the freedom to choose the one that best suits their needs. For a continuous range of predictions, the regressor model is recommended. However, the classification model takes a more hands-on approach to solving the supply-demand indicator problem mentioned earlier.

Gradient Boost Regressor Model Based:

- According to the gradient boost regressor, the storage issues reported in the last three months have the highest score (0.98), indicating the most significant influence on the product weight in tons. If the warehouse experiences a higher number of storage issues, it must possess the capability to ship a greater product weight in tons.
    - The R-Squared values is 0.994, indicating high significance.
    - The analysis reveals the importance of several factors: government certificates for warehouses, transportation issues, temporary machinery registration, and the number of retail shops. These insights can guide strategies to enhance warehouse performance.

Gradient Boost Classifier Model Based:

- According to the gradient boost classifier, the main factors determining the difference between the types are approved government certificate rating (68%), storage issues reported in l3m (18%), and product weights in tons (12%).
    - We have clustered the warehouses into three types with average recommended shipping weight:
        - Type 1: 28,927 Tons
        - Type 2: 25,350 Tons
        - Type 3: 10,501 Tons
    - Follow the average recommended product weight in tons for the respective warehouse type.
- The accuracy of the classification model is 99.6%, so there is a chance of 0.4% for misclassification, so do consider it.

Warehouse Infrastructure Based:

- 2,139 warehouses have been impacted by flooding without having any floodproofing measures in place, which are essential for preventing warehouse breakdowns.
- 65.7% of the warehouses have electricity backup, investing in electricity backup would reduce the risk of warehouse breakdowns
- Approved warehouse government certificate plays a major role in classifying the warehouse in terms of their shipping potential in product weight, as often A and A+

rated warehouses are able to deliver higher loads, so understanding the criteria for rating further would help in improving the business with its demand.

Warehouse Location Based:

- Despite the number of warehouses located in the south, the number of competitors is lower compared to that of the west. So, assuming it should be more profitable to run warehouses there, consider investing more in that zone.
- Urban warehouses encounter more storage issues, suggesting the need for targeted strategies

Warehouse Employee Based:

- Warehouses with a higher number of workers typically exhibit lower product weights, suggesting that addressing this issue could enhance efficiency.

Additional suggestion for the company:

- Consider updating the model's training dataset periodically to keep the model updated on upcoming prediction.
- Consider collecting periodic data from the warehouse, which would enable us to build more beneficial models in terms of forecasting.
- Incorporate data-sharing across different department to keep inventory and production in check.