



CAPSTONE PROJECT

Supply Chain

Submitted by:

Name: Hima Jose Paliakkara

Batch: Aug 2021-2022

Literature Review

Inventory is the stock of goods held for doing business. It can be any company's most valuable asset. It is critical for each company to manage its inventory effectively with no excess stock stored and meeting the client requirements. Inventory management is primarily specifying the placement of stocked goods. The problem we are discussing here is about the instant noodles company which is experiencing inventory cost loss due to the inadequate supply management. The management intends to maximise the amount of supply in every warehouse across the entire nation. The goal of this project is to create a model utilising past data that will establish the ideal weight of the product to be delivered to the warehouse on each occasion.

Visualization is the window to the data. It is essential in understanding the given records, attributes and the relationships. The Python programming language will be used to conduct this investigation using univariate and bivariate analysis. Dropping records containing missing values can lead to loss of vital information. Thus, they have been imputed using suitable measures. There are many possibilities for data analysis, making it challenging to choose which approach and machine learning model to employ because the model's effectiveness depends on the parameters included in the data. Data modelling is done using different regression models and tuning techniques.

The models are evaluated used accuracy score and RMSE values and chose the best model for further analysis. Important features that affect the optimum weight of the product to be shipped are Warehouse that are established atleast 5 years ago and its importance increases with the age of warehouse, Warehouse breakdown, refilling & transport issue. If business focus on these features and minimize the accidents happened and transport issues and increases the refilling time for older warehouses, based on this optimum weight to be shipped can be determined.

Table of Contents

LIST OF FIGURES.....	3
LIST OF TABLES.....	3
1. INTRODUCTION.....	4
2. EDA AND BUSINESS IMPLICATION.....	4
3. DATA CLEANING AND PRE-PROCESSING.....	13
4. MODEL BUILDING.....	15
5. MODEL VALIDATION.....	25
6. FINAL INTERPRETATION / RECOMMENDATION.....	26
APPENDIX.....	28

List of Tables

TABLE 1 EDA SUMMARY FOR CONTINUES VARIABLE SUMMARY	7
TABLE 2 EDA SUMMARY FOR CATEGORICAL/NOMINAL VARIABLE	9
TABLE 3 VIF FOR SELECTED VARIABLES	16
TABLE 4 BASE MODEL RESULTS	18
TABLE 5 ENSEMBLE MODEL RESULTS	20
TABLE 6 HYPERTUNED MODELS RESULTS AFTER HYPERTUNING	24
TABLE 7 MODEL COMPARISON.....	25

List of Figures

FIGURE 1 DISTRIBUTION OF PRODUCT DEMANDS(AVERAGES) ACROSS GEOGRAPHY	10
FIGURE 2 PRODUCT DEMAND VS OWNERSHIP AND LOCATION TYPE	10
FIGURE 3 TRANSPORT ISSUES ACROSS ZONES AND OWNERSHIP.....	10
FIGURE 4 FLOOD PROOFING ACROSS ZONES	11
FIGURE 5 <i>WAREHOUSE ELECTRICITY AND TEMPERATURE REGULATION</i>	11
FIGURE 6 PRODUCT DEMAND VS CERTIFICATIONS	11
FIGURE 7 HEATMAP.....	12
FIGURE 8 FRACTION OF MISSING VALUES(LEFT) & AFTER KNN IMPUTATION(RIGHT)	13
FIGURE 9 BOXPLOT WITH OUTLIERS.....	14
FIGURE 10 VARIABLES BOXPLOT AFTER OUTLIER TREATMENT	14
FIGURE 11 AGEGROUP DISTRIBUTION	15
FIGURE 12 FEATURE IMPORTANCE OF RANDOM FOREST.....	26
FIGURE 13 STORAGE ISSUES VS PRODUCT(LEFT) & REFILL REQUESTS VS PRODUCT (RIGHT).....	30
FIGURE 14 NUMBER OF RETAIL SHOPS VS PRODUCT(LEFT) & COMPETITION VS PRODUCT (RIGHT)	30
FIGURE 15 WORKER STRENGTH VS PRODUCT (LEFT) & GOVERNMENT CHECKS VS PRODUCT (RIGHT)	31
FIGURE 16 WAREHOUSE BREAKDOWN VS PRODUCT (LEFT) & TRANSPORT ISSUES VS PRODUCT (RIGHT)	31

1.INTRODUCTION

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. The objective of this exercise is to build a model, using historical data that will determine an optimum weight of the product to be shipped each time to the warehouse. (Dataset & Data Dictionary Refer Appendix)

Purpose of Study

The company is facing inventory cost loss due to the poor supply management of the instant noodles. The management wants to optimize the supply quantity in each and every warehouse in entire country. The objective of this project is to build a model, using historical data that will determine an optimum weight of the product to be shipped each time to the warehouse. In this problem PORDUCT_WG_TON is the target variable. With a lot of options available to analyse data, it is very difficult to decide which method and machine learning model to use since the performance of the model vary on the parameters available in the data.

This project aims to compare different popular machine learning classifiers, and measure their performance to find out which machine learning model performs better. This exploration will be carried out using the Python programming language through univariate and bivariate analysis. This exploration will be carried out using the Python programming language through univariate and bivariate analysis. The presence of outliers interferes with the correct modelling and interpretation of the data. Thus, the outliers (extreme values) are identified and replaced. Since the dataset used is related to supply chain important parameters are identified and the machine learning models are trained with the dataset for detection of the optimum weight. The study helps to analyse/optimize the supply quantity at each warehouse in the country & thereby determining the advertising strategies & campaigns for specific pockets.

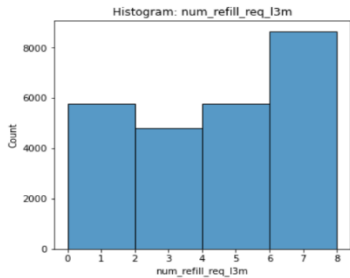
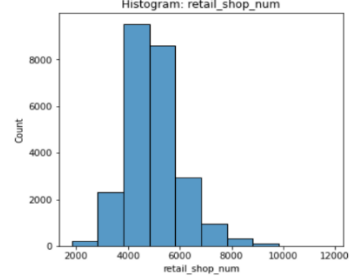
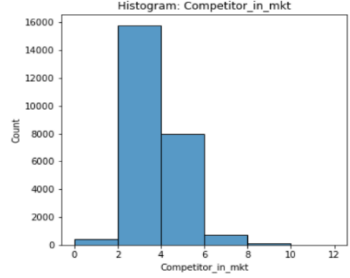
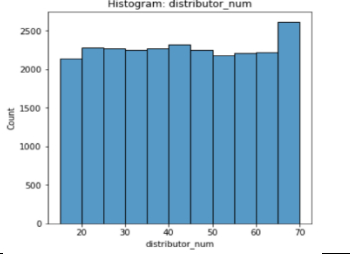
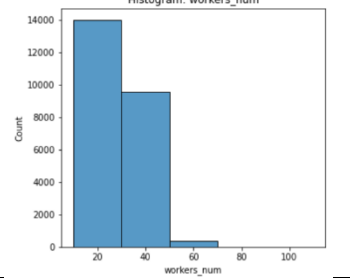
2. EDA AND BUSINESS IMPLICATION

Dataset Summary

(Pls refer Summary & Info table in Appendix 1 & 2)

- The dataset contains 25000 rows and 24 Columns
- 6 Categorical Variables, 4 Nominal Variables & 14 Continuous Variables
- The product_wg_ton is the target variable, that is the objective of the study is to study and model the product weight in order to estimate the future demands.
- No duplicates present in the dataset
- Most of the variables are right skewed, left skewed variables are: wh_breakdown_13m, num_refill_req_13m, govt_check_13m, electric_supply (refer Appendix 3)

Continuous Variables

Variable Name	Plot	Observation
num_refill_req_13m	 <p>Histogram: num_refill_req_13m</p> <p>The histogram shows the count of refills requested in the last 3 months. The x-axis represents the number of refills (0 to 8), and the y-axis represents the count (0 to 8000). The distribution is skewed to the right, with the highest frequency (over 8000) occurring at 7 refills.</p>	A fair majority has had 4 to 8 refills in the past 3 months
retail_shop_num	 <p>Histogram: retail_shop_num</p> <p>The histogram shows the count of retail shops. The x-axis represents the number of retail shops (2000 to 12000), and the y-axis represents the count (0 to 8000). The distribution is right-skewed, with the highest frequency (around 9000) occurring between 4000 and 6000 shops.</p>	Retail shops are right skewed. Having 4000 to 5000 retail shops in normal
Competitor_in_mkt	 <p>Histogram: Competitor_in_mkt</p> <p>The histogram shows the count of competitors in the market. The x-axis represents the number of competitors (0 to 12), and the y-axis represents the count (0 to 16000). The distribution is right-skewed, with the highest frequency (around 15000) occurring at 2 competitors.</p>	competitors in the market are right skewed. Having 2 to 3 competitors in norm
distributor_num	 <p>Histogram: distributor_num</p> <p>The histogram shows the count of distributors. The x-axis represents the number of distributors (20 to 70), and the y-axis represents the count (0 to 2500). The distribution is fairly uniform across the range, with counts generally between 2000 and 2500.</p>	The number of distributors is observed to be fairly uniform across the respective ranges
workers_num	 <p>Histogram: workers_num</p> <p>The histogram shows the count of workers in warehouses. The x-axis represents the number of workers (20 to 100), and the y-axis represents the count (0 to 14000). The distribution is right-skewed, with the highest frequency (around 14000) occurring at 20 workers.</p>	The majority of warehouses have 20 to 40 workers

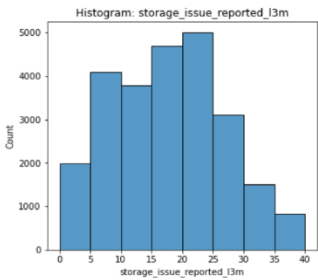
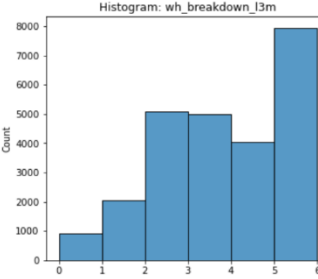
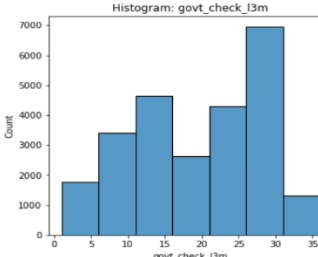
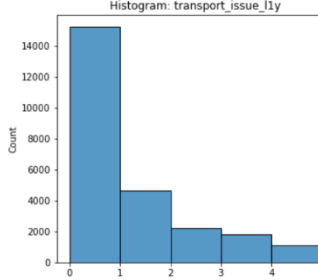
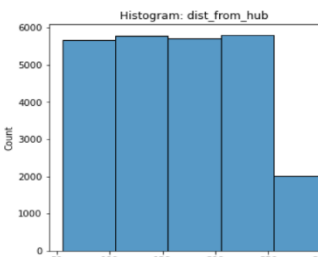
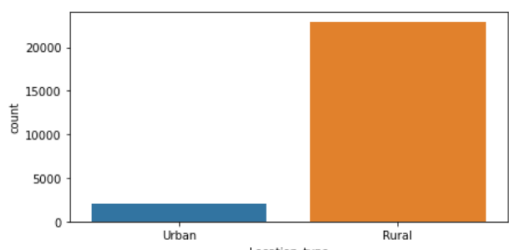
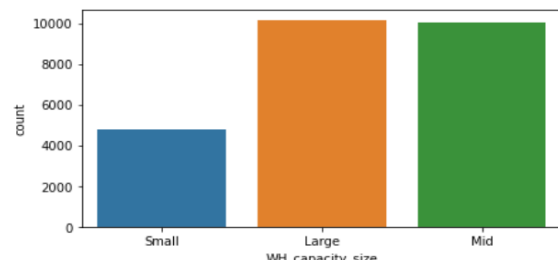
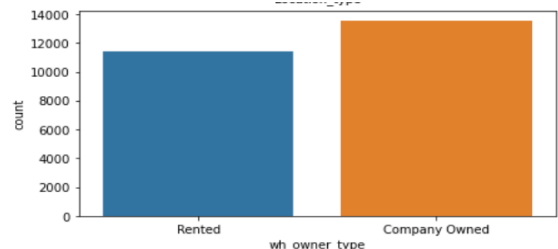
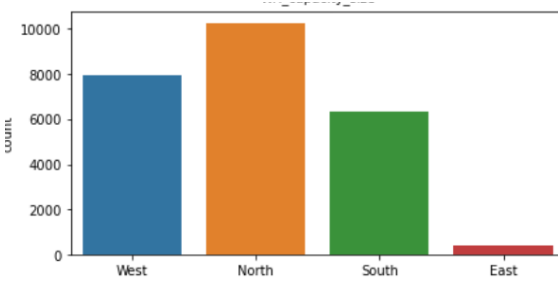
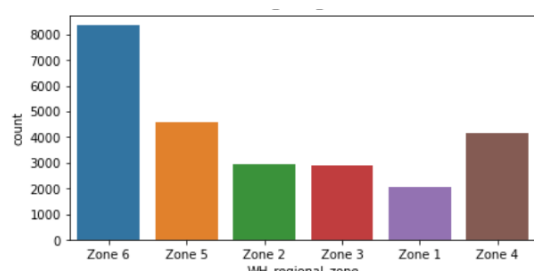
storage_issue_reported_13m	 <p>Histogram: storage_issue_reported_13m</p>	The number of storage issues reported is right skewed, with the majority having less than 20 issues across the duration of 3 months. The bin width is set to 10
wh_breakdown_13m	 <p>Histogram: wh_breakdown_13m</p>	It is observed that the majority of warehouses have reported more than 4 breakdowns in past 3 months. The data is left skewed.
govt_check_13m	 <p>Histogram: govt_check_13m</p>	The number of government checks have been 25 to 30 for most warehouses.
transport_issue_11y	 <p>Histogram: transport_issue_11y</p>	Transport issue reported are comparably less
dist_from_hub	 <p>Histogram: dist_from_hub</p>	distance from hub are observed to be fairly uniform across the respective ranges

Table 1 EDA Summary for Continues variable Summary

Categorical/Nominal Variables

Variable Name	Plot	Observation														
Location_type	 <table><thead><tr><th>Location_type</th><th>count</th></tr></thead><tbody><tr><td>Urban</td><td>~2,000</td></tr><tr><td>Rural</td><td>~22,000</td></tr></tbody></table>	Location_type	count	Urban	~2,000	Rural	~22,000	Type of Location is Urban and Rural, with a higher count in rural type								
Location_type	count															
Urban	~2,000															
Rural	~22,000															
WH_capacity_size	 <table><thead><tr><th>WH capacity size</th><th>count</th></tr></thead><tbody><tr><td>Small</td><td>~4,500</td></tr><tr><td>Large</td><td>~10,000</td></tr><tr><td>Mid</td><td>~10,000</td></tr></tbody></table>	WH capacity size	count	Small	~4,500	Large	~10,000	Mid	~10,000	Warehouse capacity are of 3 types: Small, Mid & large, it can be observed that small warehouses are less compared to that of Large and Mid.						
WH capacity size	count															
Small	~4,500															
Large	~10,000															
Mid	~10,000															
wh_owner_type	 <table><thead><tr><th>wh_owner_type</th><th>count</th></tr></thead><tbody><tr><td>Rented</td><td>~11,500</td></tr><tr><td>Company Owned</td><td>~13,500</td></tr></tbody></table>	wh_owner_type	count	Rented	~11,500	Company Owned	~13,500	There are 2 Owner types: Rented & company owner. Company Owned warehouses are slightly more than the rented ones								
wh_owner_type	count															
Rented	~11,500															
Company Owned	~13,500															
zone	 <table><thead><tr><th>zone</th><th>count</th></tr></thead><tbody><tr><td>West</td><td>~8,000</td></tr><tr><td>North</td><td>~10,000</td></tr><tr><td>South</td><td>~6,500</td></tr><tr><td>East</td><td>~500</td></tr></tbody></table>	zone	count	West	~8,000	North	~10,000	South	~6,500	East	~500	The count of warehouses in the East zone are minimal compared to those in the North, West and South zones.				
zone	count															
West	~8,000															
North	~10,000															
South	~6,500															
East	~500															
WH_regional_zone	 <table><thead><tr><th>WH regional zone</th><th>count</th></tr></thead><tbody><tr><td>Zone 6</td><td>~8,500</td></tr><tr><td>Zone 5</td><td>~4,500</td></tr><tr><td>Zone 2</td><td>~3,000</td></tr><tr><td>Zone 3</td><td>~3,000</td></tr><tr><td>Zone 1</td><td>~2,000</td></tr><tr><td>Zone 4</td><td>~4,000</td></tr></tbody></table>	WH regional zone	count	Zone 6	~8,500	Zone 5	~4,500	Zone 2	~3,000	Zone 3	~3,000	Zone 1	~2,000	Zone 4	~4,000	regional zone numbered as Zone1 to Zone6, with Zone 1 has the lowest
WH regional zone	count															
Zone 6	~8,500															
Zone 5	~4,500															
Zone 2	~3,000															
Zone 3	~3,000															
Zone 1	~2,000															
Zone 4	~4,000															

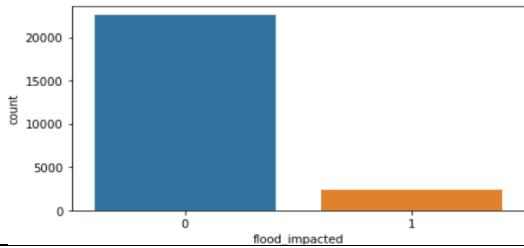
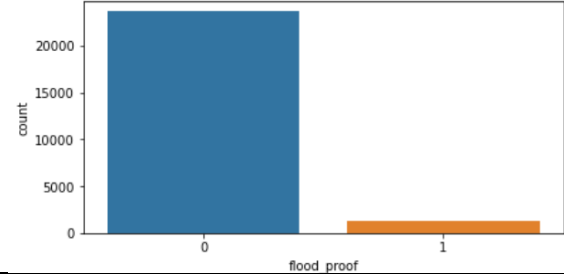
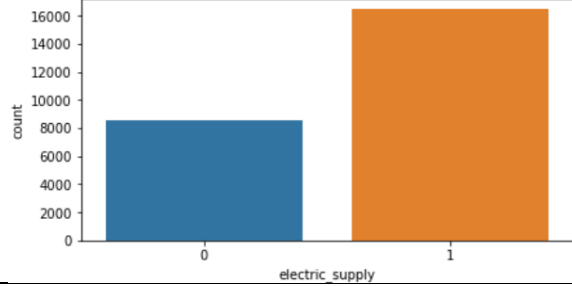
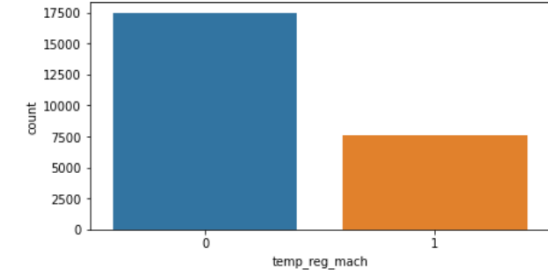
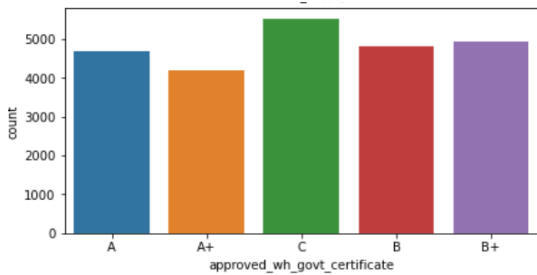
flood_impacted		It can be observed that the warehouses impacted by floods form a mere minority
flood_proof		Warehouses impacted by floods are minority. This justifies why warehouses have not opted for flood proofing
electric_supply		We can see that most of the warehouses run with electricity back up
temp_reg_mach		majority of warehouses with electric supply have opted to go ahead without regulation of temperature facility.
Approve_wh_govt_certificate		The warehouse also has a certificate (A+, A, B, B+ and C) based on the government standards.

Table 2 EDA Summary for Categorical/Nominal variable

Bivariate/Multivariate Analysis

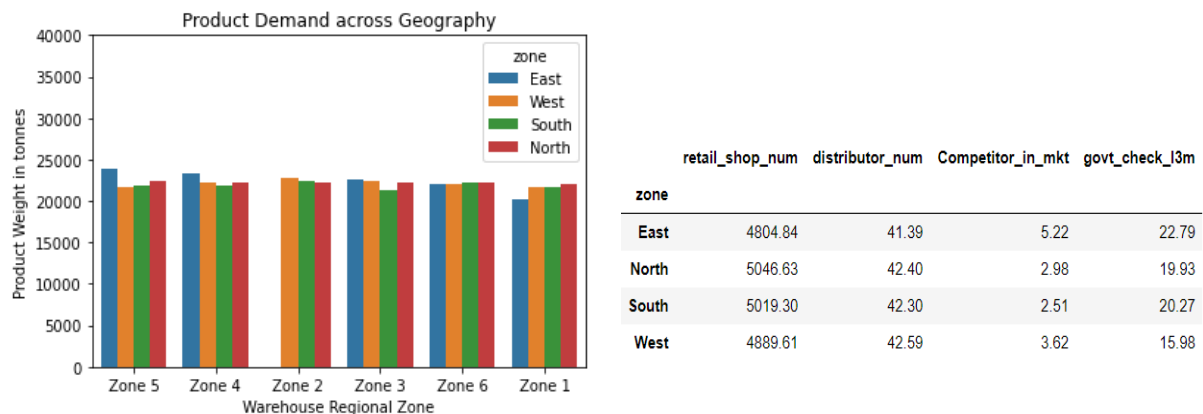


Figure 1 Distribution of product demands(averages) across geography

The East zone has no warehouses in Zone2 as can be seen. In the North and the West, the Zone6 has the highest number of warehouses. Further investigation shows that the East zone has fewer number of distributors and retail shops, but more competitors in the market, relative to the other zones. In summary, this zone has fewer warehouses, retail outlets and distributors. It has much higher number of competitors yet has the same product demand as other zones. This might be a result of the popularity of the product in this region, encouraging the marketing department to pay greater attention to this region.

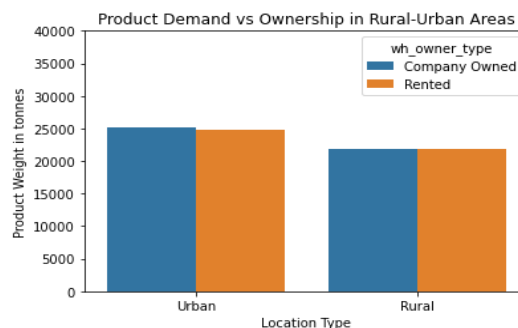


Figure 2 Product Demand vs Ownership and Location Type

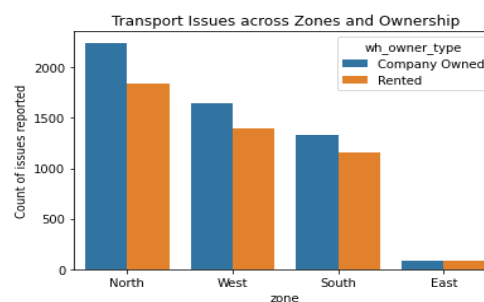


Figure 3 Transport Issues across Zones and Ownership

The ownership of the warehouse seems to have an impact on the average product weight shipped, which also varies based on the location of the warehouse. The urban company-owned warehouses order more than the rented ones in the same location. In rural areas, this difference is negligible. A greater number of transport issues have been reported by company-owned warehouses, in comparison to the rented ones. This trend is consistent in all zones, except the East, where there is no difference.

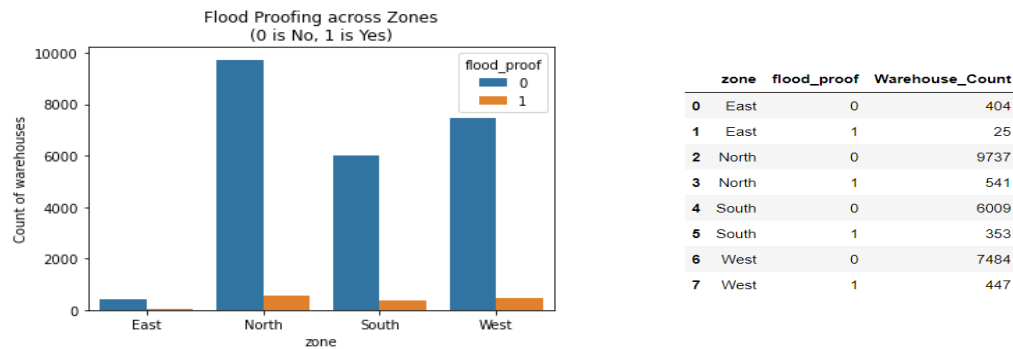


Figure 4 Flood Proofing across Zones

Flood proofing though a desirable facility at any storage facility, it can be observed that the warehouses impacted by floods form a mere minority. This justifies why warehouses have not opted for flood proofing

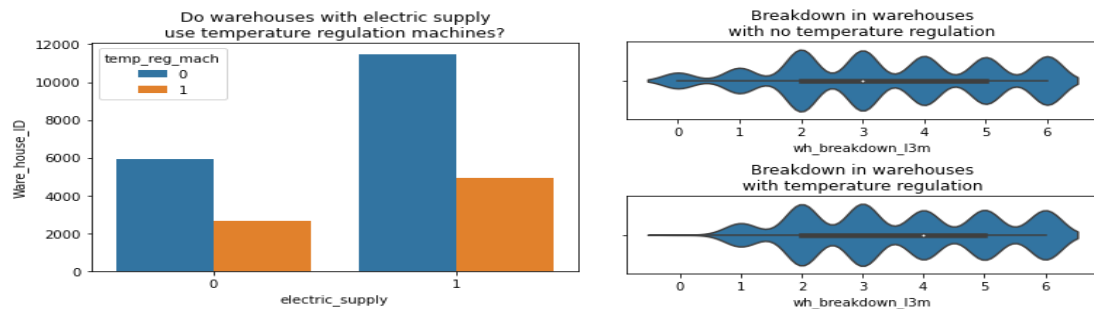


Figure 5 Warehouse Electricity and Temperature Regulation

Though the regulation of temperature can be useful in minimizing spoilage due to environmental causes, it is observed that the majority of warehouses with electric supply have opted to go ahead without this facility. The warehouses with temperature regulation, however, report more breakdowns than those without, as seen in the above graph (right). Breakdowns lead to monetary losses, and measures to minimize such incidents should be taken.

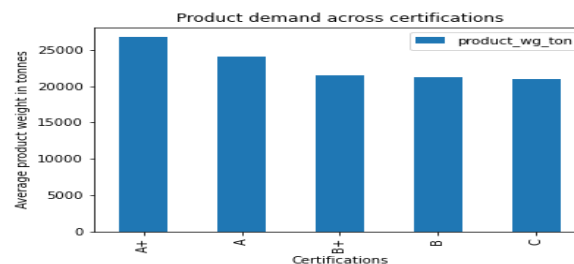


Figure 6 Product Demand vs Certifications

As seen from above, the warehouses given A+ certifications have on average greater demand for products.

Further Bivariate plots can refer on Appendix 4

Correlation Heatmap

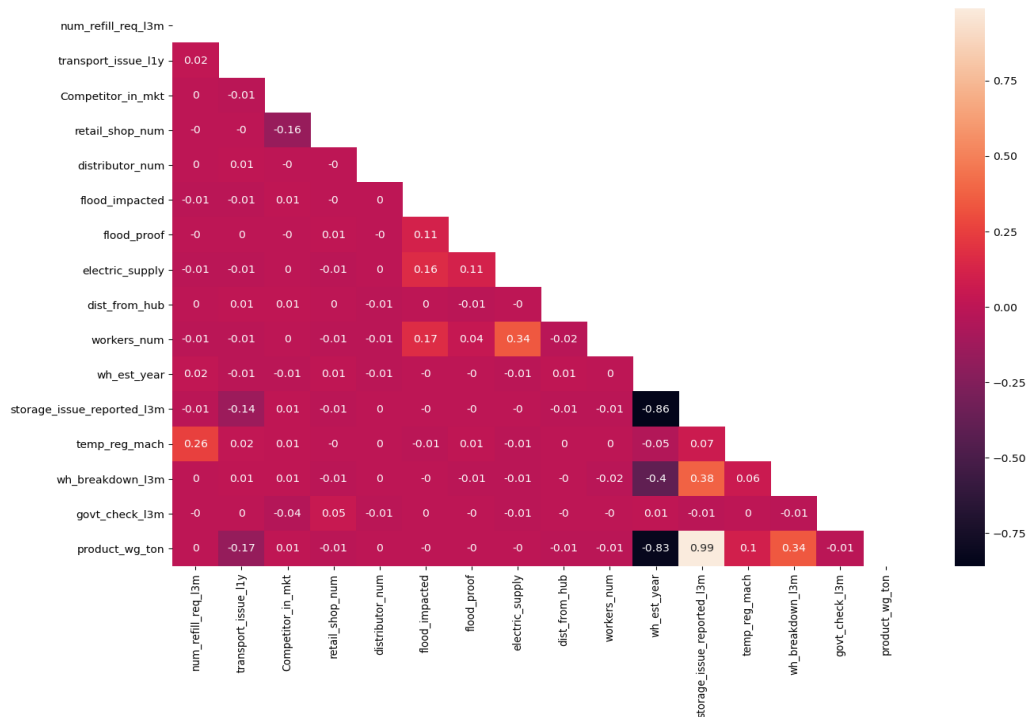


Figure 7 Heatmap

It is observed that the product demand has a strong positive correlation with the storage issues reported. The Pearson correlation coefficients calculated for all pairs of continuous variables obtained as a matrix, have been presented as a heatmap. The darker the colour, the lower is the magnitude of the correlation. As can be observed, “storage issues reported in 3 months” column has a high correlation with product weight in tons.

Business Insights from EDA

- More warehouses are situated in the rural regions, than urban
- There are unusually low number of warehouses in the East region. This region has more competition and more scope of growth. Thus, the marketing team can focus on this region. The legal team needs to figure out why this place gets more government checks than other regions.
- One third of the warehouses have no electricity
- A minority have opted for flood proofing and temperature regulation. Those who have a temperature regulator report more breakdowns than those without, so the quality of these machines should be improved
- The warehouses given A+ certifications have on average greater demand for products.

- A greater number of transport issues have been reported by company-owned warehouses, in comparison to the rented ones. This trend is consistent in all zones, except the East, where there is no difference
- The ownership of the warehouse seems to have an impact on the average product weight shipped, which also varies based on the location of the warehouse. The urban company-owned warehouses order more than the rented ones in the same location.

3. DATA CLEANING AND PRE-PROCESSING

Missing Value Treatment

There are missing values present in the dataset. Percent of Total Missing Values in the data = 2.3 % (shown below left)

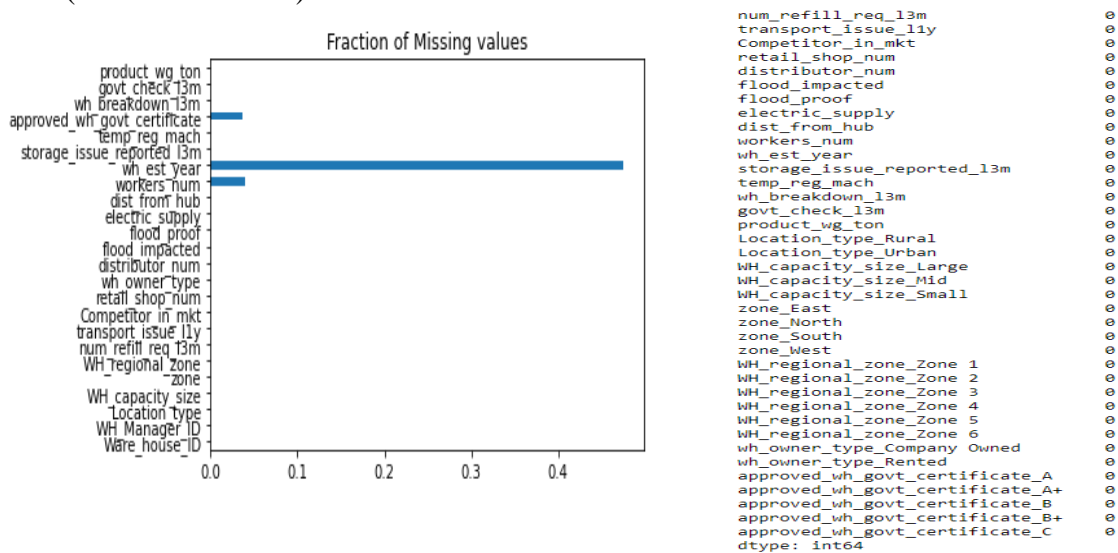


Figure 8 Fraction of missing values(left) & After KNN imputation(right)

The imputation of data employs the KNN imputer. The column listing the government certifications, being categorical in nature, are label encoded and then imputed using KNN. The number of nearest neighbours to be considered is kept at 1. However, as they are categorical in nature, these are converted to dummy variables eventually. The output after the KNN imputation is shown above (right side)

Outlier Treatment

The descriptive analysis of the data has been followed by outlier detection and removal. This is limited to continuous variables. Below are the variables having outliers

- Competitor_in_mkt, retail_shop_num, transport_issue_11y, workers_num, the outlier treatment is done in the further analysis.
- Flood_impacted & flood_proof outliers are not treated as they are nominal columns.

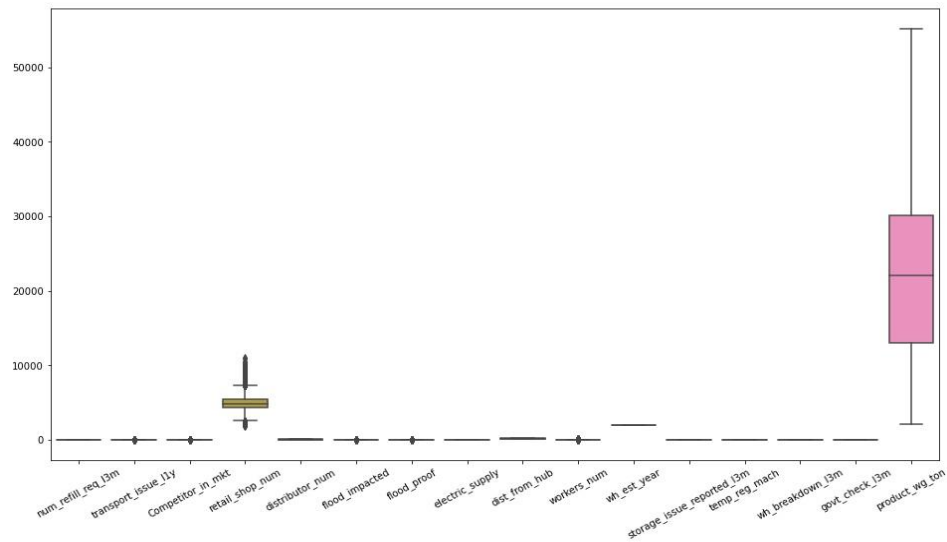


Figure 9 Boxplot with Outliers

The IQR (Interquartile Range) method has been used to detect and remove outliers. IQR is the range between the first and the third quartiles namely $Q1$ and $Q3$: $IQR = Q3 - Q1$. The data points which fall below $Q1 - 1.5 IQR$ or above $Q3 + 1.5 IQR$ are outliers. For the treatment of such outliers, the values lesser than lower bound are replaced with the value of lower bound, and the values greater than upper bound are replaced by the value of upper bound. Thus, the range of the values shrinks

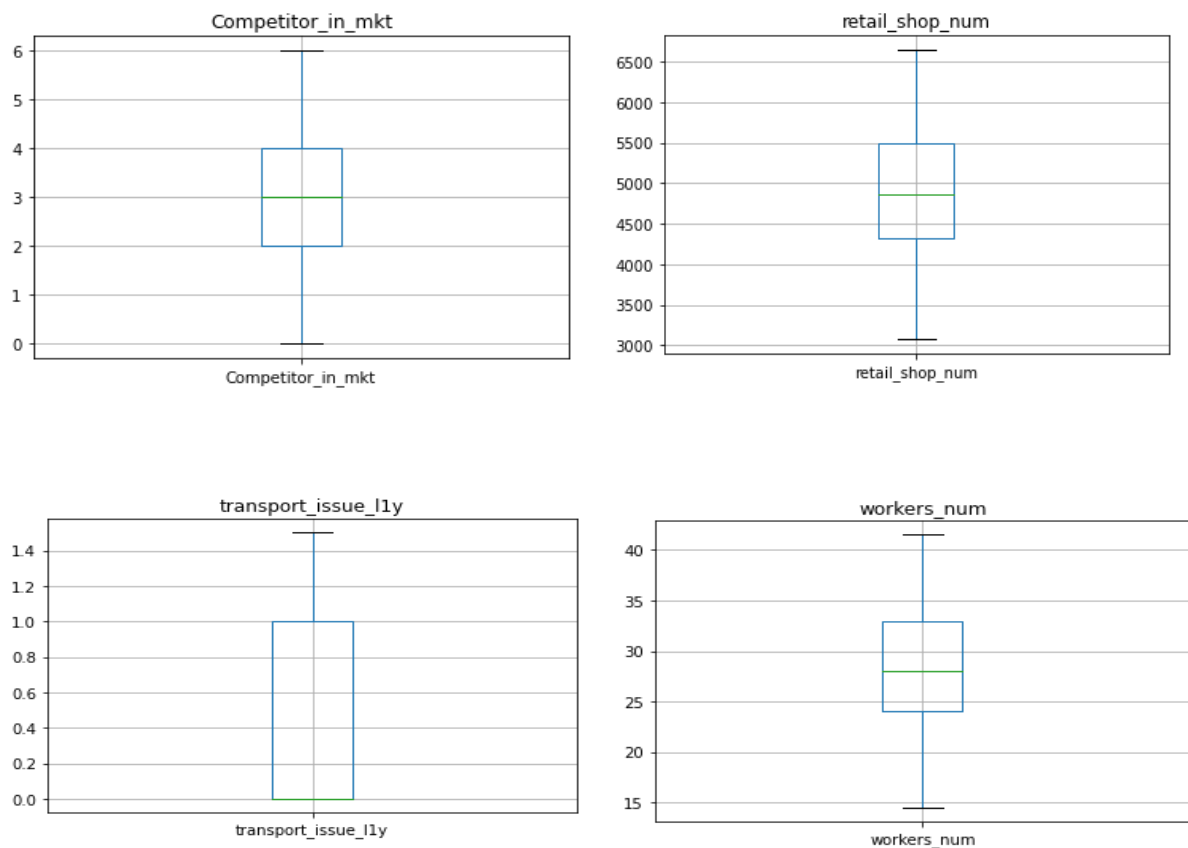


Figure 10 Variables Boxplot after outlier treatment

Removal of unwanted variables: There are two ID variables: Warehouse ID and Warehouse Manager ID. These columns have not been included in this analysis.

Creation of new Variable: Created new variable name: **AgeGroup** with binning of year into 5 sections instead of variable wh_est_year with year of warehouse established. After the creation of variable, I have dropped the variable wh_est_year. Since this variable (Agegroup) is categorical, did a get_dummies for this variable before splitting the data.

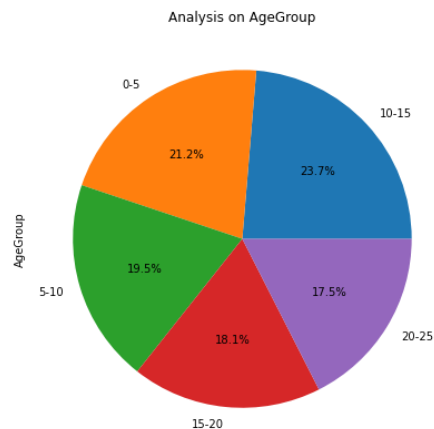


Figure 11 Agegroup Distribution

Data Imbalance and its Treatment

Data imbalance is applicable to the classification problem. Since this is a Regression problem, we no need do any data Imbalance treatment here. Clustering is done for unsupervised data, when no historical data is given. Here in our study, we are given with the historical data and it is a supervised problem, hence clustering is not applicable.

4. MODEL BUILDING

Feature Selection

Feature Selection is the method of reducing the input variable to your model by using only relevant data and getting rid of noise in data. Feature selection can be done in multiple ways but there are broadly 3 categories:

1. Filter Method
2. Wrapper Method
3. Embedded Method

1.Filter Method

In this method you filter and take only the subset of the relevant features. The model is built after selecting the features. The filtering here is done using correlation matrix and it is most commonly done using Pearson correlation and VIF

Variance Inflation Factor (VIF)

The Variance Inflation Factor (VIF) measures the severity of multicollinearity in regression analysis. Collinearity is the state where two variables are highly correlated and contain similar information about the variance within a given dataset.

Here we perform the VIF and will remove the variables one by one which are highly correlated and proceeding with the other variable for modelling. We set the threshold to 10, as we wish to remove the variable for which the remaining variables explain more than 90% of the variation. One can choose the threshold other than 10. (it depends on the business requirements). Initial VIF values for all variables (Refer Appendix 5) After few iterations the VIF, **below are the variable selected to build the model.**

	VIF_Factor	Features
0	9.58	Competitor_in_mkt
1	7.27	distributor_num
2	7.15	dist_from_hub
3	6.23	wh_breakdown_I3m
4	5.99	govt_check_I3m
5	4.82	WH_regional_zone_Zone_6
6	3.97	WH_regional_zone_Zone_5
7	3.84	WH_regional_zone_Zone_4
8	3.69	num_refill_req_I3m
9	3.07	electric_supply
10	3.01	WH_regional_zone_Zone_2
11	2.96	WH_regional_zone_Zone_3
12	2.26	WH_capacity_size_Small
13	2.25	AgeGroup_10to15
14	2.17	approved_wh_govt_certificate_Aplus
15	1.99	AgeGroup_15to20
16	1.98	AgeGroup_20to25
17	1.98	zone_West
18	1.98	temp_reg_mach
19	1.96	approved_wh_govt_certificate_C
20	1.96	AgeGroup_5to10
21	1.92	wh_owner_type_Rented
22	1.88	approved_wh_govt_certificate_Bplus
23	1.86	approved_wh_govt_certificate_B
24	1.81	zone_South
25	1.62	transport_issue_I1y
26	1.16	flood_impacted
27	1.10	Location_type_Urban
28	1.08	flood_proof

Table 3 VIF for selected variables

Train – Test Split

After selecting the variable for model building, we performed the train test split.

- X= Copy all the predictor variables & y= target into the y dataframe().
- Splitting the X and y into training and test set in 70:30 ratio with random_state=1

The dimension of X_train is (17500, 29)

The dimension of X_test is (7500, 29)

Scaling

Data standardization is the process where using which we bring all the data under the same scale. Here, we are building a model, to predict optimum weight of the product to be shipped each time to the warehouse. In this case we are expected to build model using LinearRegression, LDA, Ridge, Lasso, ANN etc. So, we are scaling the data (x_train_scaled,x_test_scaled) and will use this scaled data to perform the models where scaling is necessary.

Models

Since this is a supervised regression problem we will be performing some of the regression models below. Two metrics that statisticians often use to quantify how well a model fits a dataset are the root mean squared error (RMSE) and the R-squared (R^2).

Base models used for model building

Linear Regression: Linear Regression is the supervised Machine Learning model in which the model finds the best fit linear line between the independent and dependent variable. It is mostly used for finding out the relationship between variables and forecasting.

Test results:

	RMSE	Accuracy Score
Train	6309.61	0.71
Test	6116.42	0.72

Coefficients of Linear Regression model (refer Appendix 6)

Lasso Regression: Lasso regression is like linear regression; Linear regression gives you regression coefficients as observed in the dataset. The lasso regression allows you to shrink or regularize these coefficients to avoid overfitting and make them work better on different datasets

Test results:

	RMSE	Accuracy Score
Train	6309.61	0.71
Test	6116.42	0.72

Coefficients of Lasso Regression model (refer Appendix 7)

Ridge Regression: Ridge regressor is basically a regularized version of a Linear Regressor. The regularized term has the parameter 'alpha' which controls the regularization of the model.

Test Results:

	RMSE	Accuracy Score
Train	6309.61	0.71
Test	6116.42	0.72

Coefficients of Ridge Regression model (refer Appendix 8)

Decision Tree Regressor: Decision tree builds regression models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.

Test Results:

	RMSE	Accuracy Score
Train	0.00	1.00
Test	8279.36	0.48

Important features of Decision tree Regressor model (refer Appendix 9)

Random forest regressor: A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

Test Results:

	RMSE	Accuracy Score
Train	2260.83	0.96
Test	5941.91	0.73

Important features of Random Forest Regressor model (refer Appendix 10)

MLP regressor: A multilayer perceptron (MLP) is a fully connected class of feedforward artificial neural network (ANN).

Test Results:

	RMSE	Accuracy Score
Train	6270.82	0.71
Test	6080.12	0.72

After running the base models below are the R^2 score and RMSE values:

Models	Train RMSE	Test RMSE	Training Score	Test Score
Ridge Regression	6309.62	6113.95	0.71	0.72
Lasso Regression	6309.63	6113.68	0.71	0.72
Linear Regression	6309.61	6113.80	0.71	0.72
Decision Tree Regressor	0.00	8160.90	1.00	0.50
Random Forest Regressor	2257.56	5941.59	0.96	0.73
ANN Regressor	5946.95	5838.63	0.74	0.74

Table 4 Base Model Results

Insights:

- We see the results are almost similar for linear, lasso and ridge regression. Since the features are selected using VIF method, lasso and ridge are performing same as linear regression
- Decision tree and Random Forest's nonlinear nature gives better results than linear regression. Decision tree's accuracy shows that it is overfitting, so does random forest's results show
- Linear regression and other methods can understand only linear relationships, to understand non-linear relationships ANN works better. Looking at the result ANN performs better than Linear and regularization methods. Real life data is supposed to have complex non-linear relationships, that's why ANN is giving better results than linear model
- From the models we could see that warehouse established year, number of refills, warehouse breakdown, distribution from hub etc are few of the importance features effecting the optimum weight shipment
- We can use grid search to tackle this problem Later, we will try to tune the models and will see whether the model performance improves

Ensemble modelling

Ensemble methods are techniques that aim at improving the accuracy of results in models by combining multiple models instead of using a single model. Tried using few of the ensemble models below to see whether the model performs better than base models.

Bagging regressor: It is an ensemble meta-estimator that fits base regressors each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction.

Test Results:

	RMSE	Accuracy Score
Train	2661.88	0.95
Test	6164.40	0.71

Important features of Bagging Regressor model (refer Appendix 11)

AdaBoost regressor: It is a meta-estimator that begins by fitting a regressor on the original dataset and then fits additional copies of the regressor on the same dataset but where the weights of instances are adjusted according to the error of the current prediction. It is best used with weak learners.

Test Results:

	RMSE	Accuracy Score
Train	6847.90	0.65
Test	6724.97	0.66

Important features of AdaBoost Regressor model (refer Appendix 12)

Gradient Boost regressor: Gradient Boosting algorithm is used to generate an ensemble model by combining the weak learners or weak predictive models.

Test Results:

	RMSE	Accuracy Score
Train	5974.91	0.74
Test	5830.55	0.74

Important features of Gradient Boost Regressor model (refer Appendix 13)

Extreme Gradient Boosting: XGBoost is a powerful approach for building supervised regression models. It is an efficient implementation of gradient boosting that can be used for regression predictive modelling.

Test results:

	RMSE	Accuracy Score
Train	4364.92	0.86
Test	6001.27	0.73

Important features of XGBoost Regressor model (refer Appendix A.14)

Ensemble Model results as below:

Models	Train RMSE	Test RMSE	Training Score	Test Score
AdaBoostRegressor	6847.90	6724.97	0.65	0.66
GradientBoostingRegressor	5974.91	5830.55	0.74	0.74
BaggingRegressor	2661.88	6164.40	0.95	0.71
XGBRegressor	4364.92	6001.27	0.86	0.73

Table 5 Ensemble Model results

Insights:

- Both Bagging & XGB regressor models are not performing well
- Within all the ensemble models shown above Gradient Boosting regressor is performing better
- From the models we could see that warehouse established year, transport issue, warehouse breakdown are few of the importance features effecting the optimum weight shipment
- Later, we will try to tune the models and will see whether the model performance improves

Model Hypertuning

Model Tuning is the process of maximizing a model's performance without overfitting or creating too high of a variance. This is accomplished by selecting appropriate "hyperparameters", these parameters are set manually. The three most commonly used approaches are **Grid Search, Random Search & K-fold**. Here I am using GridSearchCV Method for the model tuning

- Grid Search: Grid search also known as parameter sweeping. This method involves manually defining a subset of the hyperparametric space and exhausting all combinations of the specified hyperparameter subsets. (refer Appendix 15)

Ridge Regressor with GridSearchCV

The given tuning parameters are below:

```
alpha : (np.linspace(0.1, 1, 25)),
solver: ['svd', 'cholesky', 'sag', 'saga', 'lsqr', 'lbfgs', 'sparse_cg'],
tol    : [0.001, 0.1]}
CV     : 10
```

The best parameters after fitting the model are:

```
{'alpha': 0.1, 'solver': 'saga', 'tol': 0.1}
```

Test Results after Tuning:

	RMSE	Accuracy Score
Train	6522.99	0.69
Test	6366.76	0.70

Lasso Regressor with GridSearchCV

The given tuning parameters are below:

```
alpha : (np.linspace(0.1, 1, 25))
tol    : [0.0001, 0.001, 0.1]
CV     : 10
```

The best parameters after fitting the model are:

```
{'alpha': 0.16875, 'tol': 0.0001}
```

Test Results after Tuning:

	RMSE	Accuracy Score
Train	6310.90	0.71
Test	6115.72	0.72

DecisionTree Regressor with GridSearchCV

Building a Decision Tree Regressor using a grid search cross validation to get best parameter/estimators for a given dataset. The given tuning parameters are below:

```
GridSearchCV(cv=3, estimator=DecisionTreeRegressor(random_state=1),
             param_grid={'max_depth': [20, 25, 30, 35, 40, 50],
                          'min_samples_leaf': [3, 15, 18, 30],
                          'min_samples_split': [15, 30, 35, 40, 50]})
```

The best parameters after fitting the model are:

```
{'max_depth': 20, 'min_samples_leaf': 30, 'min_samples_split': 15}
```

Test Results after Tuning:

	RMSE	Accuracy Score
Train	5529.03	0.77
Test	5944.87	0.73

Random Forest Regressor with GridSearchCV

The given tuning parameters are below:

```
GridSearchCV(cv=5, estimator=RandomForestRegressor(random_state=1),  
             param_grid={'max_depth': [10, 15, 20], 'max_features': [4, 6, 8],  
                          'min_samples_leaf': [5, 15, 30],  
                          'min_samples_split': [20, 30, 50],  
                          'n_estimators': [300, 400]})
```

The best parameters after fitting the model are:

```
{'max_depth': 20, 'max_features': 8, 'min_samples_leaf': 5, 'min_samples_split': 20, 'n_estimators': 300}
```

Test Results after Tuning:

	RMSE	Accuracy Score
Train	4998.51	0.82
Test	5753.39	0.75

NN MLP regressor using GridSearchCV

The given tuning parameters are below:

```
GridSearchCV(cv=3, estimator=MLPRegressor(max_iter=500, random_state=1),  
             param_grid={'activation': ['tanh', 'relu'],  
                          'hidden_layer_sizes': [500, (100, 100)],  
                          'solver': ['sgd', 'adam']})
```

The best parameters after fitting the model are:

```
{'activation': 'relu', 'hidden_layer_sizes': 500, 'solver': 'adam'}
```

Test Results after Tuning:

	RMSE	Accuracy Score
Train	6270.85	0.71
Test	6080.12	0.72

Adaboost Reressor with GridSearchCV

The given tuning parameters are below:

```
GridSearchCV(cv=3, estimator=AdaBoostRegressor(), n_jobs=-1,
             param_grid={'learning_rate': [0.01, 0.05, 0.1, 1],
                          'loss': ['linear', 'square', 'exponential'],
                          'n_estimators': array([10, 20, 30, 40, 50, 60, 70, 80, 90])})
```

The best parameters after fitting the model are:

```
{'learning_rate': 0.1, 'loss': 'linear', 'n_estimators': 30}
```

Test Results after Tuning:

	RMSE	Accuracy Score
Train	6702.77	0.67
Test	6521.56	0.68

GradientBossting with GridSreachCV

The given tuning parameters are below:

```
params_GBR_GS = {"max_depth": [3,5,6,7],
                  "min_samples_split": [2, 3, 10],
                  "min_samples_leaf": [1, 3, 10],
                  'learning_rate':[0.05,0.1,0.2],
                  'n_estimators': [10,20,30]}
```

The best parameters after fitting the model are:

```
{'learning_rate': 0.1, 'max_depth': 7, 'min_samples_leaf': 10,
 'min_samples_split': 2, 'n_estimators': 30}
```

Test Results after Tuning:

	RMSE	Accuracy Score
Train	5536.09	0.77
Test	5733.53	0.75

Bagging with GridSerachCV

The given tuning parameters are below:

```
params_bag_GS = {"n_estimators": [200,300],
                  "max_features":[20,30,50],
                  "max_samples": [0.5,0.1,1],
                  "bootstrap": [True, False],
                  "bootstrap_features": [True, False]}
```

The best parameters after fitting the model are:

```
{'bootstrap': True, 'bootstrap_features': False,
 'max_features': 20, 'max_samples': 0.1, 'n_estimators': 200}
```

Test Results after Tuning:

	RMSE	Accuracy Score
Train	6289.33	0.71
Test	6810.36	0.65

XGB Regressor with GridSearchCV

The given tuning parameters are below:

```
params_xgbR_GS = {"max_depth": [3,4,5,6,7],
                  "min_child_weight" : [4,5,6,8],
                  'learning_rate':[0.05,0.1,0.2,0.25,0.8,1],
                  'n_estimators': [30,50,100]}
```

The best parameters after fitting the model are:

```
{'learning_rate': 0.05, 'max_depth': 7,
 'min_child_weight': 8, 'n_estimators': 100}
```

Test Results after Tuning:

	RMSE	Accuracy Score
Train	5634.44	0.79
Test	5724.93	0.75

Models	Train RMSE	Test RMSE	Training Score	Test Score
Ridge Regression with GridSreachCV	6522.99	6366.76	0.69	0.69
Lasso Regression with GridSreachCV	6310.90	6115.72	0.71	0.72
Linear Regression with GridSreachCV	6309.61	6116.42	0.71	0.72
Decision Tree Regressor with GridSreachCV	5529.03	5944.87	0.77	0.73
Random Forest Regressor with GridSreachCV	4998.51	5753.39	0.82	0.75
ANN Regressor with GridSreachCV	6270.82	6080.12	0.71	0.72
AdaBoostRegressor with GridSreachCV	6702.77	6521.56	0.67	0.68
GradientBoostingRegressor with GridSreachCV	5536.09	5733.53	0.77	0.75
BaggingRegressor with GridSreachCV	6289.33	6810.36	0.71	0.65
XGBRegressor with GridSreachCV	5364.44	5724.93	0.79	0.75

Table 6 Hypertuned models results after hypertuning

Insights:

- We can find that Ridge, Lasso is not showing much improvement in the score and RMSE after tuning and results of both are almost same.
- Decision Tree regressor has improved after performing model tuning
- ANN regressor is giving the same values even after performing the hypertuning
- We can find that Gradient boost regressors has improved
- Among all the models performed till now, XGBRegressor & Random Forest are performing well

5. MODEL VALIDATION

Two metrics there are use to validate how well a model fits a dataset are the root mean squared error (RMSE) and the R-squared (R^2), which are calculated as follows:

- **RMSE:** A metric that tells us how far apart the predicted values are from the observed values in a dataset, on average. The lower the RMSE, the better a model fits a dataset
- **R^2 :** A metric that tells us the proportion of the variance in the response variable of a regression model that can be explained by the predictor variables. This value ranges from 0 to 1. The higher the R^2 value, the better a model fits a dataset.

Final Model Comparison

Models	Train RMSE	Test RMSE	Training Score	Test Score
Ridge Regression	6309.62	6113.95	0.71	0.72
Lasso Regression	6309.63	6113.68	0.71	0.72
Linear Regression	6309.61	6113.80	0.71	0.72
Decision Tree Regressor	0.00	8160.90	1.00	0.50
Random Forest Regressor	2257.56	5941.59	0.96	0.73
ANN Regressor	5946.95	5838.63	0.74	0.74
AdaBoostRegressor	6847.90	6724.97	0.65	0.66
GradientBoostingRegressor	5974.91	5830.55	0.74	0.74
BaggingRegressor	2661.88	6164.40	0.95	0.71
XGBRegressor	4364.92	6001.27	0.86	0.73
Ridge Regression with GridSreachCV	6522.99	6366.76	0.69	0.69
Lasso Regression with GridSreachCV	6310.90	6115.72	0.71	0.72
Linear Regression with GridSreachCV	6309.61	6116.42	0.71	0.72
Decision Tree Regressor with GridSreachCV	5529.03	5944.87	0.77	0.73
Random Forest Regressor with GridSreachCV	4998.51	5753.39	0.82	0.75
ANN Regressor with GridSreachCV	6270.82	6080.12	0.71	0.72
AdaBoostRegressor with GridSreachCV	6702.77	6521.56	0.67	0.68
GradientBoostingRegressor with GridSreachCV	5536.09	5733.53	0.77	0.75
BaggingRegressor with GridSreachCV	6289.33	6810.36	0.71	0.65
XGBRegressor with GridSreachCV	5364.44	5724.93	0.79	0.75

Table 7 Model Comparison

Comparing all the model performed, I got the best results for Random Forest regressor. Random Forest regressor is giving 82% accuracy In Train & 75% in Test with RMSE value of Train as 4998.51 & Train as 5753.39

Feature Importance

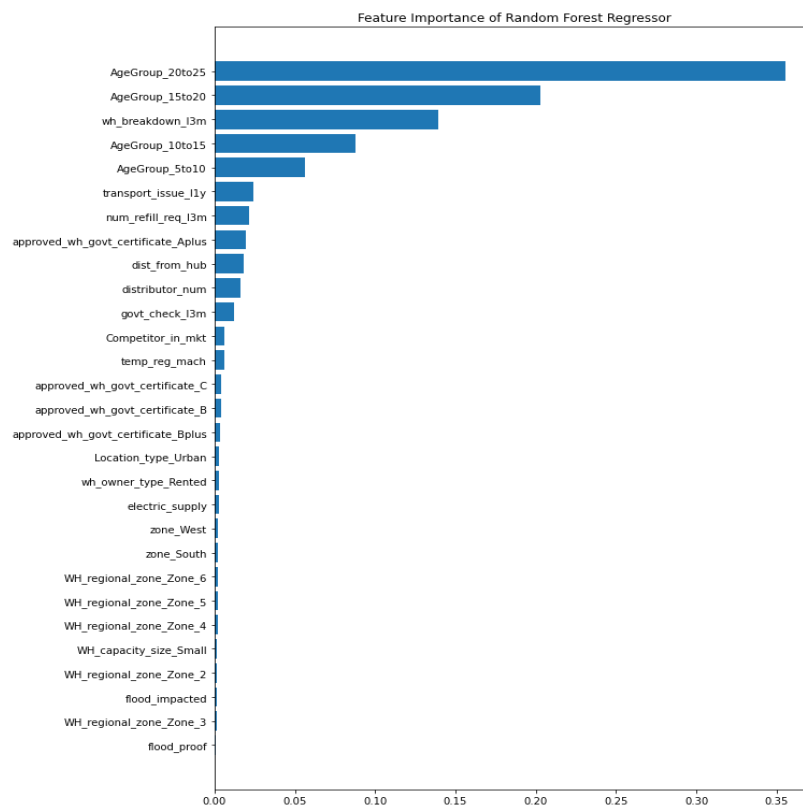


Figure 12 Feature Importance of Random Forest

These features have maximum effect on optimum weight (product_wg_ton):

- Warehouse Breakdown in last 3 months
- Transport issue
- Refilling time in last 3 months
- Warehouse established year

6. FINAL INTERPRETATION / RECOMMENDATION

- Since this is regression problem, we have tried out different regression models to confirm which performs well and gives the best accuracy
- To handle overfitting, we performed hyperparameter tuning using GridsearchCV.
- Comparing all models, we have obtained the best results for Random Forest regressor
→ Random Forest regressor is giving 82% accuracy In Train & 75% in Test with RMSE value of Train as 4998.51 & Train as 5753.39
- The warehouse breakdowns due to both internal & external factors results on its inventory management & leading manufactures.

- Accident or Product stolen shall be another factor which can lead to optimum weight mismatch at the time of delivery, resulting in supply constraints
- Delay in stock refilling hampers reduced stock during high demand times
- Features that affect product_wg_ton which specifies the optimum weight of the product to be shipped are Warehouse that are established at least 5 years ago and its importance increases with the age of warehouse.

Recommendations

- Set up a governing council that offers a clear strategy for functionality and efficiency, thus reducing Warehouse breakdown factors. The council's aim is to give directions and align the supply chain strategy with the company's core goals. The council helps in removing barriers within the organization.
- Review policies and procedures to ensure efficiency and compliance. It also helps avoid bottlenecks in the supply chain, streamline operations and mitigate the risks of theft and fraud. Regular reviews help in identifying different risk elements and estimating their financial impact.
- Include demand planning and forecasting to improve refilling. Incorporate warehouse operations that are proficient in managing inventory and accurate inventory records. This helps to know whenever there is a refill required immediately and won't affect supply.
- Need to perform frequent audits upon warehouse operation standards. Use technology to improve the supply chain. Review all the existing processes that are affecting the inventory management. Determine the areas where implementing technology could improve the processes. The right strategies have the potential to transform your supply chain and increase revenue
- East zone has fewer warehouses, retail outlets and distributors. It has much higher number of competitors yet has the same product demand as other zones. This might be a result of the popularity of the product in this region, encouraging the marketing department to pay greater attention to this region.

APPENDIX

Dataset: [data](#)

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_13m	Number of times refilling has been done in last 3 months
transport_issue_1ly	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 distributor 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_13m	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_13m	Number of time warehouse face a breakdown in last 3 months. Like strike from worker, flood, or electrical failure
govt_check_13m	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

1. Data Summary

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Ware_house_ID	25000	25000	WH_118479	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
WH_Manager_ID	25000	25000	EID_61594	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Location_type	25000	2	Rural	22957	NaN	NaN	NaN	NaN	NaN	NaN	NaN
WH_capacity_size	25000	3	Large	10169	NaN	NaN	NaN	NaN	NaN	NaN	NaN
zone	25000	4	North	10278	NaN	NaN	NaN	NaN	NaN	NaN	NaN
WH_regional_zone	25000	6	Zone 6	8339	NaN	NaN	NaN	NaN	NaN	NaN	NaN
num_refill_req_l3m	25000.0	NaN	NaN	NaN	4.08904	2.606612	0.0	2.0	4.0	6.0	8.0
transport_issue_l1y	25000.0	NaN	NaN	NaN	0.77368	1.199449	0.0	0.0	0.0	1.0	5.0
Competitor_in_mkt	25000.0	NaN	NaN	NaN	3.1042	1.141663	0.0	2.0	3.0	4.0	12.0
retail_shop_num	25000.0	NaN	NaN	NaN	4985.71156	1052.825252	1821.0	4313.0	4859.0	5500.0	11008.0
wh_owner_type	25000	2	Company Owned	13578	NaN	NaN	NaN	NaN	NaN	NaN	NaN
distributor_num	25000.0	NaN	NaN	NaN	42.41812	16.064329	15.0	29.0	42.0	56.0	70.0
flood_impacted	25000.0	NaN	NaN	NaN	0.09816	0.297537	0.0	0.0	0.0	0.0	1.0
flood_proof	25000.0	NaN	NaN	NaN	0.05464	0.227281	0.0	0.0	0.0	0.0	1.0
electric_supply	25000.0	NaN	NaN	NaN	0.65688	0.474761	0.0	0.0	1.0	1.0	1.0
dist_from_hub	25000.0	NaN	NaN	NaN	163.53732	62.718609	55.0	109.0	164.0	218.0	271.0
workers_num	24010.0	NaN	NaN	NaN	28.944398	7.872534	10.0	24.0	28.0	33.0	98.0
wh_est_year	13119.0	NaN	NaN	NaN	2009.383185	7.52823	1996.0	2003.0	2009.0	2016.0	2023.0
storage_issue_reported_l3m	25000.0	NaN	NaN	NaN	17.13044	9.161108	0.0	10.0	18.0	24.0	39.0
temp_reg_mach	25000.0	NaN	NaN	NaN	0.30328	0.459684	0.0	0.0	0.0	1.0	1.0
approved_wh_govt_certificate	24092	5	C	5501	NaN	NaN	NaN	NaN	NaN	NaN	NaN
wh_breakdown_l3m	25000.0	NaN	NaN	NaN	3.48204	1.690335	0.0	2.0	3.0	5.0	6.0
govt_check_l3m	25000.0	NaN	NaN	NaN	18.81228	8.632382	1.0	11.0	21.0	26.0	32.0
product_wg_ton	25000.0	NaN	NaN	NaN	22102.63292	11607.755077	2065.0	13059.0	22101.0	30103.0	55151.0

2. Dataset Info

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25000 entries, 0 to 24999
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)
memory usage: 4.6+ MB
```

3. Skewness of variables

	Skewness
flood_proof	3.92
flood_impacted	2.70
transport_issue_l1y	1.61
workers_num	1.06
Competitor_in_mkt	0.98
retail_shop_num	0.91
temp_reg_mach	0.86
product_wg_ton	0.33
storage_issue_reported_l3m	0.11
distributor_num	0.02
wh_est_year	0.01
dist_from_hub	-0.01
wh_breakdown_l3m	-0.07
num_refill_req_l3m	-0.08
govt_check_l3m	-0.36
electric_supply	-0.66

4. Bivariate plots

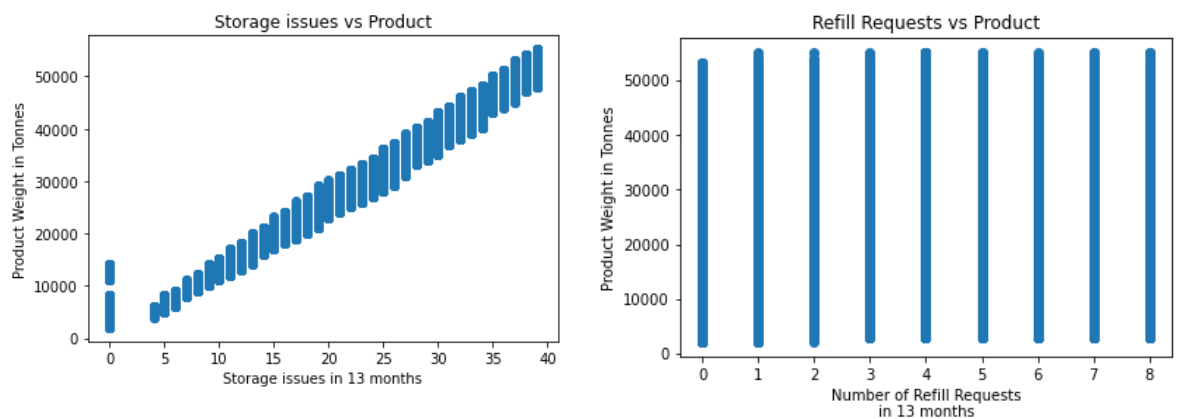


Figure 13 Storage Issues vs Product(left) & Refill Requests vs Product (Right)

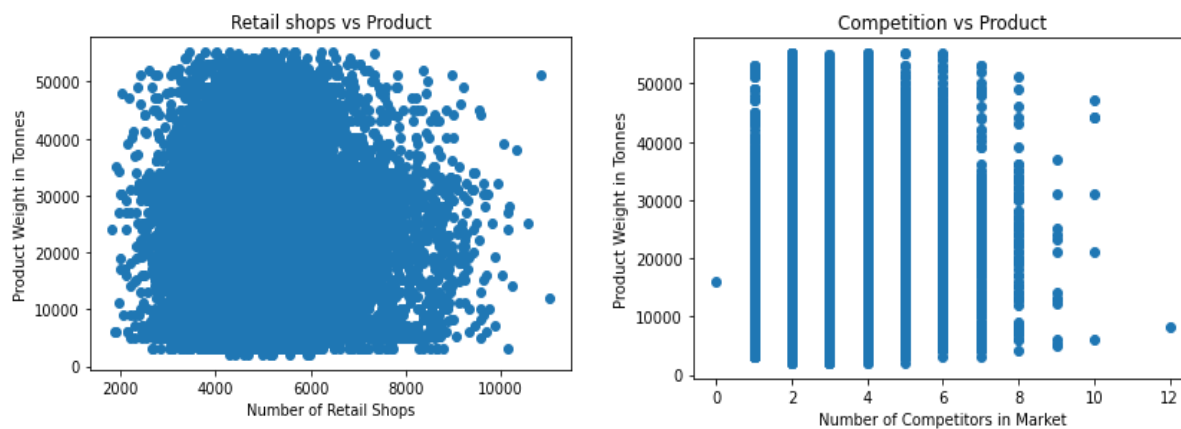


Figure 14 Number of Retail Shops vs Product(left) & Competition vs Product (Right)

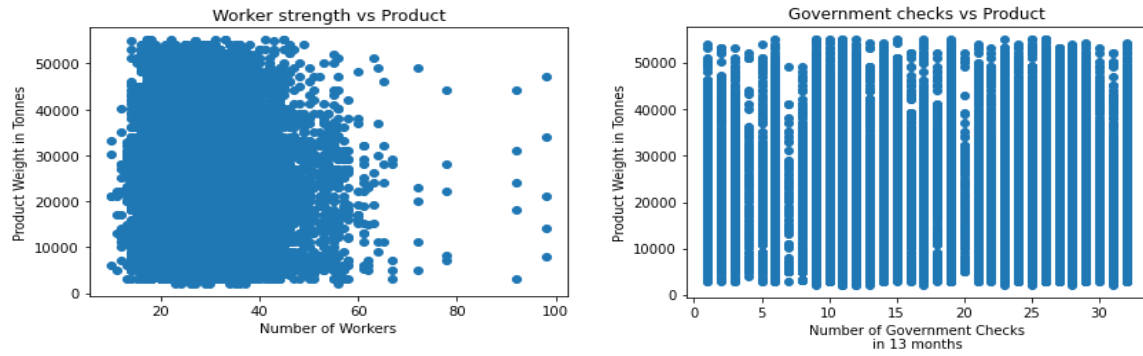


Figure 15 Worker Strength vs Product (left) & Government Checks vs Product (Right)

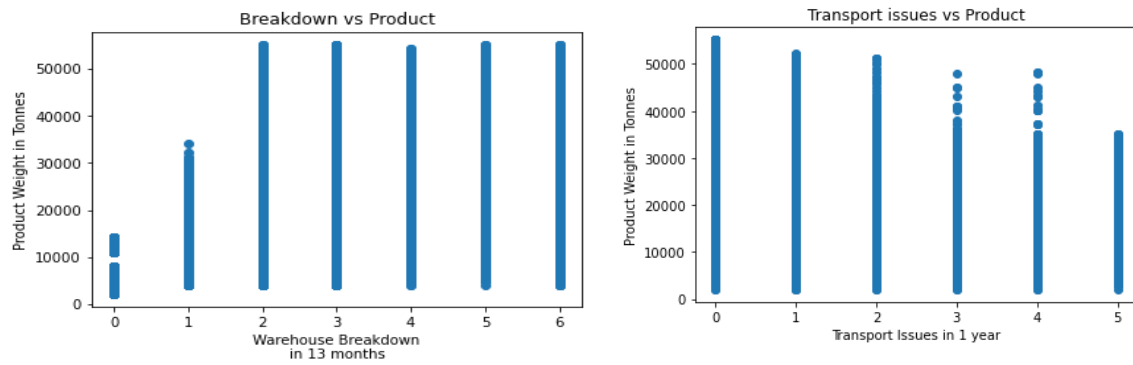
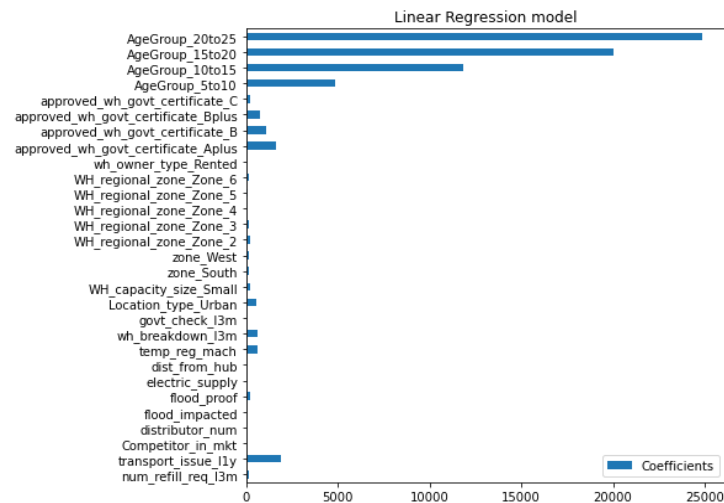


Figure 16 Warehouse Breakdown vs Product (left) & Transport Issues vs Product (Right)

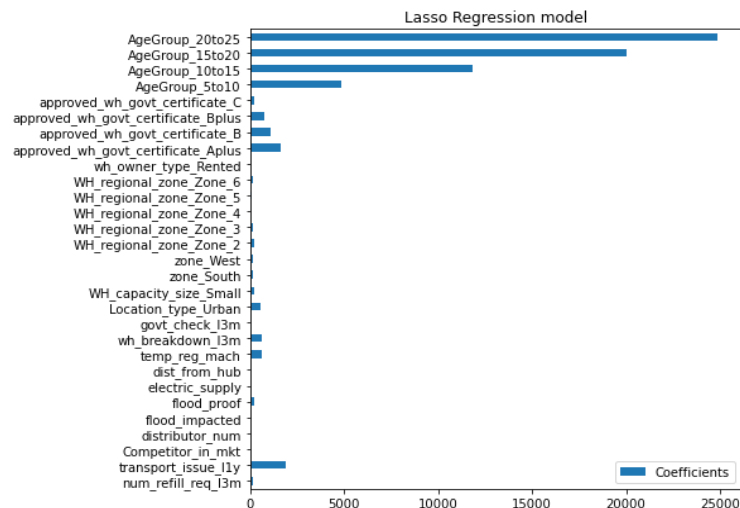
5. VIF values for all variables

VIF_Factor		Features
0	inf	WH_capacity_size_Mid
1	inf	WH_regional_zone_Zone_2
2	inf	WH_regional_zone_Zone_3
3	inf	WH_regional_zone_Zone_4
4	27.83	retail_shop_num
5	20.77	workers_num
6	17.08	zone_North
7	16.47	storage_issue_reported_I3m
8	13.19	zone_West
9	11.22	zone_South
10	9.78	Competitor_in_mkt
11	7.72	distributor_num
12	7.56	dist_from_hub
13	7.11	govt_check_I3m
14	6.61	WH_regional_zone_Zone_6
15	6.51	wh_breakdown_I3m
16	5.43	WH_regional_zone_Zone_5
17	4.95	AgeGroup_20to25
18	4.09	AgeGroup_15to20
19	3.77	num_refill_req_I3m
20	3.55	electric_supply
21	3.29	AgeGroup_10to15
22	2.82	WH_capacity_size_Small
23	2.19	approved_wh_govt_certificate_Aplus
24	2.12	AgeGroup_5to10
25	2.00	approved_wh_govt_certificate_C
26	1.98	temp_reg_mach
27	1.96	wh_owner_type_Rented
28	1.92	approved_wh_govt_certificate_Bplus
29	1.89	approved_wh_govt_certificate_B
30	1.66	transport_issue_I1y
31	1.17	flood_impacted
32	1.10	Location_type_Urban
33	1.08	flood_proof

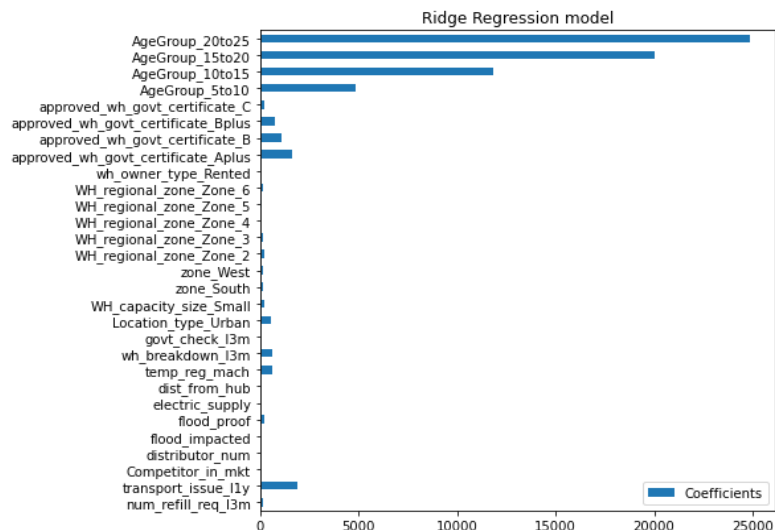
6. Coefficients of Linear Regression Base Model



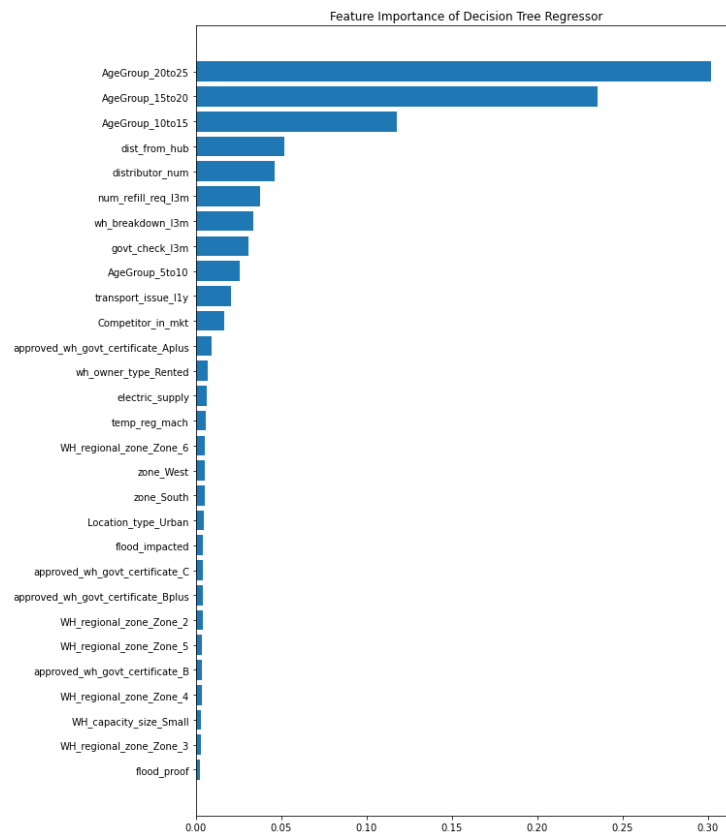
7. Coefficients of Lasso Regression Base Model



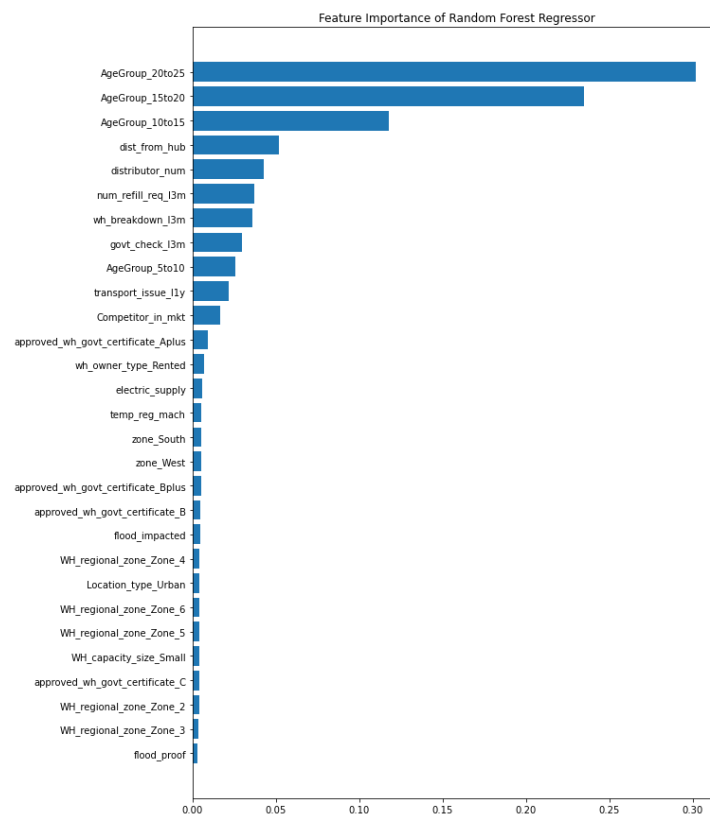
8. Coefficients of Ridge Regression Base Model



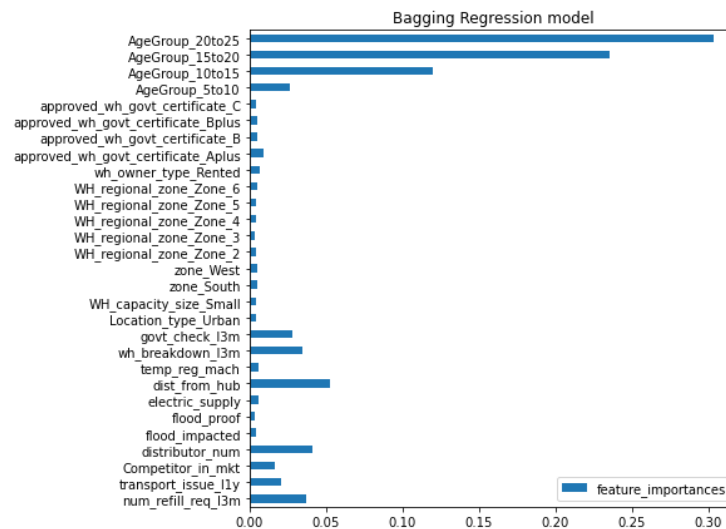
9. Important features of Decision Tree Regressor Base Model



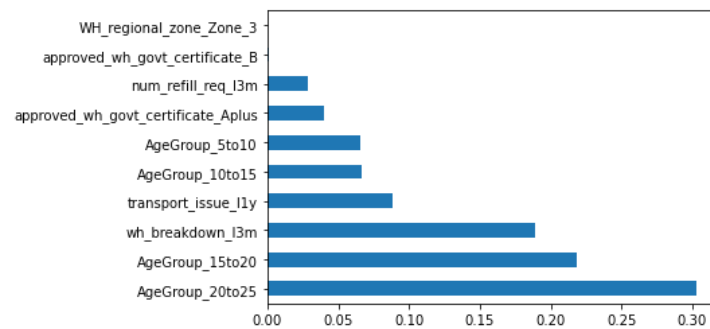
10. Important features of Random Forest Regressor Base Model



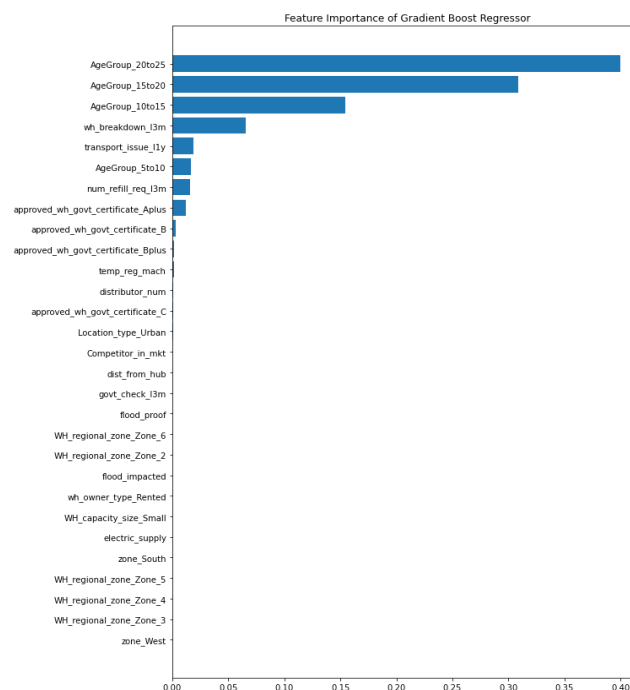
11. Feature Importance of Bagging Regressor Model



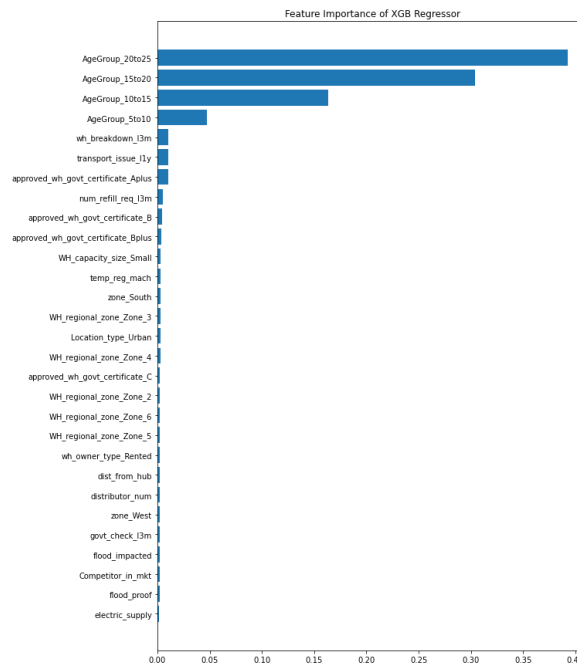
12. Important features of AdaBoost Regressor Model



13. Important features of Gradient Boost Regressor Model



14. Important features of XGB Regressor Model



15. HyperTuning Parameters

Hypertune Parameters	Description
alpha	Constant that multiplies the L2 term, controlling regularization strength. alpha must be a non-negative float
Tol	Precision of the solution
cv	the cross-validation generator or an iterable, in this case, there is a 10-fold cross validation
Solver	This parameter represents which algorithm to use in the optimization problem
Max_depth	The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples. The max_depth value should be adjusted according to high/low to avoid overfitting/underfitting
min_samples_split	int, float, optional (default=2). min_samples_split is used to control over-fitting. depending on the level of underfitting or overfitting
min_samples_leaf	int, float, optional (default=1). The minimum number of samples required to be at a leaf node. min_samples_leaf is also used to control over-fitting by defining that each leaf has more than one element.

random_state	(default=None). Random state ensures that the splits that you generate are reproducible. random state that you provide is used as a seed to the random number generator. This ensures that the random numbers are generated in the same order.
max_features	These are the maximum number of features Random Forest is allowed to try in individual tree. Increasing max_features generally improve the performance of the model as at each node now we have a higher number of options to be considered
n_estimators	This is the number of trees you want to build before taking the maximum voting or averages of predictions. Higher number of trees give you better performance but makes your code slower.
hidden_layer_sizes	tuple, length = n_layers - 2, default=(100,). The ith element represents the number of neurons in the ith hidden layer
activation	Activation functions are used to introduce nonlinearity to models, which allows deep learning models to learn nonlinear prediction boundaries.
Solver	This parameter represents which algorithm to use in the optimization problem
loss	The loss function to use when updating the weights after each boosting iteration
learning_rate	Weight applied to each regressor at each boosting iteration. A higher learning rate increases the contribution of each regressor
bootstrap	Whether samples are drawn with replacement. If False, sampling without replacement is performed
bootstrap_features	Whether features are drawn with replacement
min_child_weight	Defines the minimum sum of weights of all observations required in a child. Used to control over-fitting. Higher values prevent a model from learning relations which might be highly specific to the particular sample selected for a tree.