Capstone Report
Supply Chain Analysis

Submitted by:

N. Aishwarya
PGP-DSBA

November 2022

Business Report Outline

List of Figures

List of Tables

List of Formulas

Introduction to the Business Problem

Problem Statement:

A FMCG company has entered into the instant noodles business two years back. Their higher management has notices that there is a 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 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.

Purpose of Study:

The company is incurring inventory loss due to inadequate demand-supply management of its product, instant noodles. The management needs to optimize the supply quantity across all warehouses in the country.

Hence the objective of this project is to build a model using historical data, to determine an optimum product weight to be shipped each time to the warehouse. With multiple options available to analyze data, it is challenging to decide the appropriate 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.

Approach:

Data exploration will be carried out using the Python, 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 optimum weight. The study helps to analyze/optimize the supply quantity at each warehouse in the country & thereby determining the advertising strategies & campaigns for specific pockets.

In this problem, “PRODUCT_WG_TON is the target variable.

1. Model building and interpretation:

1.1. Build various models:

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

2. 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.

Formula 1:

$$\text{VIF} = \frac{1}{1-r^2}$$

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.

	VIF_Factor	Features
0	inf	WH_regional_zone_Zone_2
1	inf	WH_regional_zone_Zone_4
2	inf	WH_regional_zone_Zone_3
3	inf	WH_capacity_size_Mid
4	24.25	retail_shop_num
5	16.59	zone_North
6	16.48	storage_issue_reported_l3m
7	15.96	workers_num
8	12.78	zone_West
9	10.97	zone_South
10	9.27	Competitor_in_mkt
11	7.71	distributor_num
12	7.55	dist_from_hub
13	7.07	govt_check_l3m
14	6.55	WH_regional_zone_Zone_6
15	6.51	wh_breakdown_l3m
16	5.37	WH_regional_zone_Zone_5
17	4.94	AgeGroup_20to25
18	4.09	AgeGroup_15to20
19	3.77	num_refill_req_l3m
20	3.39	electric_supply

Figure 1: Variance Inflation Factor

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. Now we will remove the variables at the top "WH_regional_zone_Zone_2, retail_shop_num, workers_num, storage_issue_reported_l3m, zone_North" as its still showing VIF value greater than 10.

After few iterations the VIF, below are the variable selected to build the model.

	VIF_Factor	Features
0	8.55	Competitor_in_mkt
1	6.87	distributor_num
2	6.79	dist_from_hub
3	6.17	wh_breakdown_l3m
4	5.97	govt_check_l3m
5	3.63	num_refill_req_l3m
6	2.92	electric_supply
7	2.57	WH_regional_zone_Zone_6
8	2.24	AgeGroup_10to15
9	2.15	approved_wh_govt_certificate_Aplus
10	2.04	WH_regional_zone_Zone_4
11	2.03	WH_regional_zone_Zone_5
12	1.98	AgeGroup_15to20
13	1.98	temp_reg_mach
14	1.96	AgeGroup_20to25
15	1.96	zone_West
16	1.94	AgeGroup_5to10
17	1.93	approved_wh_govt_certificate_C
18	1.89	wh_owner_type_Rented
19	1.86	approved_wh_govt_certificate_Bplus
20	1.84	approved_wh_govt_certificate_B

Figure 2: VIF for selected variables

Now all the variables above have VIF values less than 10, will continue with these variables and so the model building.

Below is the head of the dataset after dropping the columns with VIF>10:

	num_refill_req_l3m	transport_issue_l1y	Competitor_in_mkt	distributor_num	flood_impacted	flood_proof	electric_supply	dist_from_hub	temp_reg_mach
0	3.0	1.0	2.0	24.0	0.0	0.0	1.0	91.0	0.0
1	0.0	0.0	4.0	47.0	0.0	0.0	1.0	210.0	0.0
2	1.0	0.0	4.0	64.0	0.0	0.0	0.0	161.0	0.0
3	7.0	2.5	2.0	50.0	0.0	0.0	0.0	103.0	1.0
4	3.0	1.0	2.0	42.0	0.0	0.0	1.0	112.0	0.0

Figure 3: VIF Head data for selected variables

Train -Test Split

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

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

Dimension of train and test dataset:

```
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 Linear Regression, 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 a supervised regression problem 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 (R2).

1.2. Predictive model against the test set using various appropriate performance metrics:

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.

Formula 2: Linear Regression:

$$\boldsymbol{Y = m_1X_1 + m_2X_2 + \cdots + m_nX_n + C + e}$$

Y = Dependent / target / predicted variable
X_i= Independent/ predictor variable
m_i= coefficients for the i^{th} independent / predictor variable.
C = constant / intercept / bias
e = residual error / unexplained variance / difference between actual and prediction.

Test Results:

	RMSE	Accuracy Score
Train	6307.43	0.71
Test	6110.81	0.72

Lasso Regression:

Lasso regression is similar to linear regression; Linear regression provides regression coefficients as observed in the dataset, while lasso regression allows to shrink or regularize these coefficients to avoid overfitting.

Test Results:

	RMSE	Accuracy Score
Train	6307.43	0.71
Test	6110.81	0.72

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	6307.43	0.71
Test	6110.81	0.72

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	8231.95	0.49

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	2267.59	0.96
Test	5918.08	0.74

ANN Regressor:

Artificial Neural Networks have the ability to learn the complex relationship between the features and target due to the presence of activation function in each layer.

Test Results:

	RMSE	Accuracy Score
Train	6270.43	0.71
Test	6075.72	0.72

After running the base models, below are the RMSE and R2 score values:

	Train (RMSE)	Test (RMSE)	Training (R2)	Test (R2)
Ridge Regression	6307.43	6110.81	0.71	0.72
Lasso Regression	6307.43	6110.81	0.71	0.72
Linear Regression	6307.43	6110.81	0.71	0.72
Decision Tree Regressor	0.00	8231.95	1.00	0.49
Random Forest Regressor	2267.59	5918.08	0.96	0.74
ANN Regressor	6270.43	6075.72	0.71	0.72

Table 1: Base Model Results

1.3. Interpretation of the model:

- We can observe that the results are almost similar for Linear, Lasso and Ridge regression. Hence 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 it can be inferred 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.

2. Model Tuning and business implication

2.A. 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	2684.79	0.95
Test	6150.14	0.71

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	6942.13	0.64
Test	6849.01	0.65

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	5967.89	0.74
Test	5819.52	0.74

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	4380.68	0.86
Test	5991.81	0.73

Ensemble Model results as below:

	Train RMSE	Test RMSE	Training Score	Test Score
AdaBoost Regressor	6942.13	6849.01	0.64	0.65
Gradient Boosting Regressor	5967.89	5819.52	0.74	0.74
Bagging Regressor	2684.79	6150.14	0.95	0.71
XGB Regressor	4380.68	5991.81	0.86	0.73

Table 2: 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 is 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.

2.2. Model tuning measures:

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 GridSearchCV Method is used for the model tuning.

GridSearchCV on Ridge Regression

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]}
```

The best parameters after fitting the model are:

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

Test Results after Tuning:

	RMSE	Accuracy Score
Train	6560.29	0.68
Test	6398.49	0.69

Lasso Regressor with GridSearchCV:

The given tuning parameters are below:

```
{'alpha': (np.linspace(0.05, 1 , 25)),
     'tol':[0.0001,0.001,0.1] }
```

The best parameters after fitting the model are:

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

Test Results after Tuning:

	RMSE	Accuracy Score
Train	6308.69	0.71
Test	6110.29	0.72

GridSearchCV on Decision Tree

Building a Decision Tree Regressor using a grid search cross validation to get best parameter or 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	5527.60	0.77
Test	5933.77	0.73

GridSearchCV on Random Forest Regressor:

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': 400}
```

Test Results after Tuning:

	RMSE	Accuracy Score
Train	5005.05	0.82
Test	5740.90	0.75

GridSearchCV for ANN regressor:

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': (100, 100), 'solver': 'adam'}
```

Test Results after Tuning:

	RMSE	Accuracy Score
Train	6270.43	0.71
Test	6075.72	0.72

GridSearchCV on AdaBoosting:

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': 'square', 'n_estimators': 20}
```

Test Results after Tuning:

	RMSE	Accuracy Score
Train	6717.76	0.67
Test	6535.12	0.68

GridSearchCV on Gradient Boosting:

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': 0.2,
 'max_depth': 6,
 'min_samples_leaf': 10,
 'min_samples_split': 2,
 'n_estimators': 20}
```

Test Results after Tuning:

	RMSE	Accuracy Score
Train	5535.96	0.77
Test	5730.60	0.75

Bagging with GridSearchCV:

The given tuning parameters are below:

```
params_bag_GS = {"n_estimators": [200,300], #50,100
              "max_features":[20,30,50], #12468
              "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': 0.5,
 'n_estimators': 300}
```

Test Results after Tuning:

	RMSE	Accuracy Score
Train	6185.21	0.72
Test	6707.60	0.66

GridSearchCV on XGB Regressor:

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': 6,
 'min_child_weight': 5,
 'n_estimators': 100}
```

Test Results after Tuning:

	RMSE	Accuracy Score
Train	5370.65	0.79
Test	5719.79	0.75

	Train RMSE	Test RMSE	Training Score	Test Score
Ridge Regression with GridSearchCV	6560.29	6398.49	0.68	0.69
Lasso Regression with GridSearchCV	6308.69	6110.29	0.71	0.72
Linear Regression with GridSearchCV	6307.43	6110.81	0.71	0.72
Decision Tree Regressor with GridSearchCV	5527.60	5933.77	0.77	0.73
Random Forest Regressor with GridSearchCV	5005.05	5740.90	0.82	0.75
ANN Regressor with GridSearchCV	6270.43	6075.72	0.71	0.72
AdaBoost Regressor with GridSearchCV	6717.76	6535.12	0.67	0.68
Gradient BoostingR egressor with GridSearchCV	5535.96	5730.60	0.77	0.75
Bagging Regressor with GridSearchCV	6185.21	6707.60	0.72	0.66
XGB Regressor with GridSearchCV	5370.65	5719.79	0.79	0.75

Table 3: Hypertuned models results after hypertuning

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 (R2), 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.

Formula 3:

$$RMSE=\sqrt{\frac{(e_1^2+e_2^2+\cdots+e_n^2}{n}}$$

Where $e_i = y_i - \hat{y}_i$

- **R-Square(R2):** 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 R2 value, the better a model fits a dataset.

Formula 4:

$$R^2 = 1 - \frac{SSE}{SST}$$

Where, $SST = \sum^n (Y_i - \bar{Y})^2$ and $SSE = \sum^n (Y_i - \hat{Y}_i)^2$

Final Model Comparison:

	Train RMSE	Test RMSE	Training Score	Test Score
Ridge Regression	6307.43	6110.81	0.71	0.72
Lasso Regression	6307.43	6110.81	0.71	0.72
Linear Regression	6307.43	6110.81	0.71	0.72
Decision Tree Regressor	0.00	8231.95	1.00	0.49
Random Forest Regressor	2267.59	5918.08	0.96	0.74
ANN Regressor	6270.43	6075.72	0.71	0.72
AdaBoost Regressor	6942.13	6849.01	0.64	0.65
Gradient Boosting Regressor	5967.89	5819.52	0.74	0.74
Bagging Regressor	2684.79	6150.14	0.95	0.71
XGB Regressor	4380.68	5991.81	0.86	0.73
Ridge Regression with GridSearch CV	6560.29	6398.49	0.68	0.69
Lasso Regression with GridSearchCV	6308.69	6110.29	0.71	0.72
Linear Regression with GridSearchCV	6307.43	6110.81	0.71	0.72
Decision Tree Regressor with GridSearchCV	5527.60	5933.77	0.77	0.73
Random Forest Regressor with GridSearchCV	5005.05	5740.90	0.82	0.75
ANN Regressor with GridSearchCV	6270.43	6075.72	0.71	0.72

AdaBoost Regressor with GridSearchCV	6717.76	6535.12	0.67	0.68
Gradient BoostingRegressor with GridSearchCV	5535.96	5730.60	0.77	0.75
Bagging Regressor with GridSearchCV	6185.21	6707.60	0.72	0.66
XGB Regressor with GridSearchCV	5370.65	5719.79	0.79	0.75

Table 4: Final Model Comparison

Comparing all the model performed, the most optimal model is Random Forest regressor. Random Forest regressor is giving 82% accuracy In Train & 75% in Test with RMSE value of Train as 5005.05 & Train as 5740.90.

Feature Importance of Random Forest:

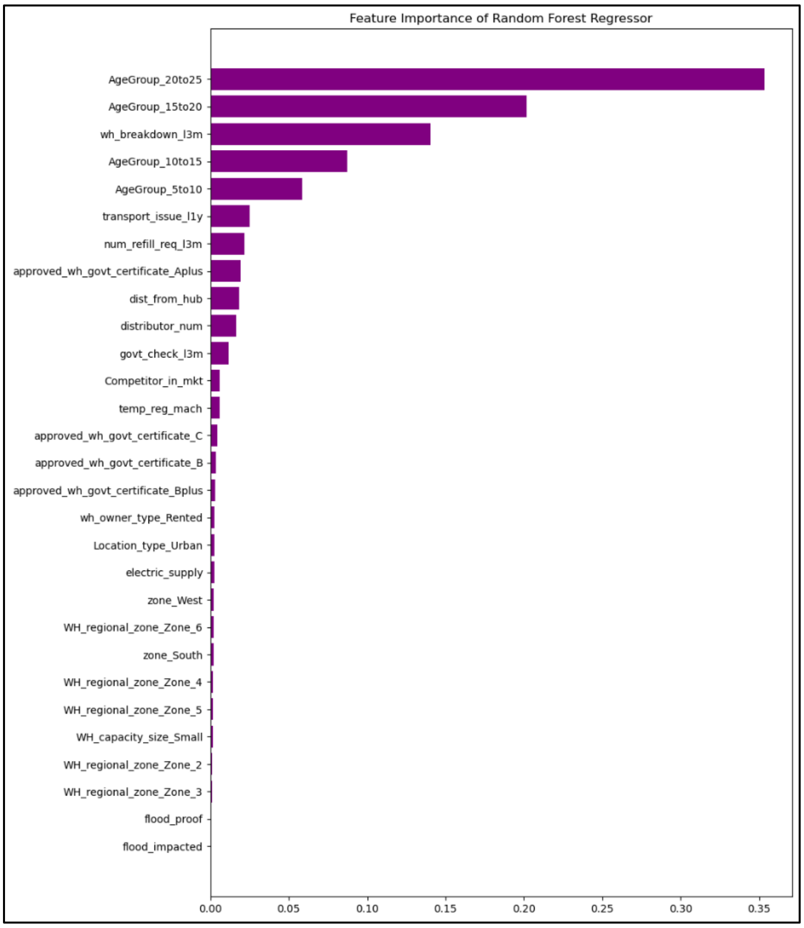

Figure 4: Feature Importance of Random Forest

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

1. Warehouse Breakdown in last 3 months.
2. Transport issue.
3. Refilling time in last 3 months.
4. Warehouse established year.

2.C. Interpretation of the most optimum model and its implication on the business

Optimal model:

- Among all the models performed till now, XGB Regressor & Random Forest are performing well.
- 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 observe that Gradient boost regressors has improved.

Model Insights:

- 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 records 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.

Implication on the business:

Based on the model outputs, the business should strive to improve across Strategic, Operational and tactical dimensions. These improvements have the potential to transform business, by supply chain being a revenue lever, beyond just cost savings.

1.Strategic dimension:

A. Improved strategic KPIs:

- Strategic KPIs such as Customer Service Level and improved fill rates needs to be improved using better demand planning and forecasting. Warehouse operations needs to be aligned to deliver inventory management and accurate inventory records. This helps to know whenever there is a refill required immediately and won’t affect supply.

B. Improved strategic focus:

- East zone's product demand is on par with other zones, however with increased competition. However, the zone has fewer warehouses, retail outlets and distributors. Hence it is critical to focus on improving more focus on this zone, by enhancing the product fulfillment experience.

2.Operational dimension:

A. Warehouse efficiency:

- 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.
- 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.

B. Operational streamlining:

- 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.

3.Tactical dimension:

A. Distribution:

- Distribution from hub is a key feature identified in impacting the business – hence improved logistics scheduling can play a key role for improvement in this area.

Appendix

Data dictionary as described below:

S. No	Field Name	Description
1	Ware_house_ID	Product warehouse ID
2	WH_Manager_ID	Employee ID of warehouse manager
3	Location_type	Location of warehouse like in city or village
4	WH_capacity_size	Storage capacity size of the warehouse
5	zone	Zone of the warehouse
6	WH_regional_zone	Regional zone of the warehouse under each zone
7	num_refill_req_l3m	Number of times refilling has been done in last 3 months
8	transport_issue_l1y	Any transport issue like accident or goods stolen reported in last one year
9	Competitor_in_mkt	Number of instant noodles competitor in the market
10	retail_shop_num	Number of retails shop who sell the product under the warehouse area
11	wh_owner_type	Company is owning the warehouse or they have got the warehouse on rent
12	distributor_num	Number of distributers works in between warehouse and retail shops
13	flood_impacted	Warehouse is in the Flood impacted area indicator
14	flood_proof	Warehouse is flood proof indicators. Like storage is at some height not directly on the ground
15	electric_supply	Warehouse have electric back up like generator, so they can run the warehouse in load shedding
16	dist_from_hub	Distance between warehouse to the production hub in Kms
17	workers_num	Number of workers working in the warehouse
18	wh_est_year	Warehouse established year
19	storage_issue_reported_l3m	Warehouse reported storage issue to corporate office in last 3 months. Like rat, fungus because of moisture etc.
20	temp_reg_mach	Warehouse have temperature regulating machine indicator
21	approved_wh_govt_certificate	What kind of standard certificate has been issued to the warehouse from government regulatory body
22	wh_breakdown_l3m	Number of time warehouse face a breakdown in last 3 months. Like strike from worker, flood, or electrical failure
23	govt_check_l3m	Number of time government Officers have been visited the warehouse to check the quality and expire of stored food in last 3 months
24	product_wg_ton	Product has been shipped in last 3 months. Weight is in tons