INDIAN STATISTICAL INSTITUTE

PROJECT REPORT

OPTIMIZING SHIPMENT WEIGHTS IN FMCG SUPPLY CHAIN

Submitted by:

Belal Ahmed Siddiqui

ROLL NO. QR2304

M. TECH IN QUALITY, RELIABILITY & OPERATIONS RESEARCH
(2ND SEMESTER, 1ST YEAR)

Under the guidance of

DR. PRASUN DAS

SQC & OR UNIT

INDIAN STATISTICAL INSTITUTE

KOLKATA

INDIAN STATISTICAL INSTITUTE

SQC & OR UNIT, KOLKATA

CERTIFICATION

This is to certify that the project report entitled *"***OPTIMIZING SHIPMENT WEIGHTS IN FMCG SUPPLY CHAIN***"* has been prepared by ***Belal Ahmed Siddiqui (QR2304), student of M.Tech QROR*** in the duration of May'24 – July '24. This serves as a part of the necessary and partial requirements for receiving the degree of M.Tech in Quality Reliability & Operations Research (QROR) awarded by the Indian Statistical Institute, Kolkata. He has carried out this work under my supervision.

Date: July 23, 2024

[Signature]

PROF. PRASUN DAS

SQC & OR UNIT

INDIAN STATISTICAL INSTITUTE

KOLKATA

ACKNOWLEDGEMENT

I would like to express my best regards and heartiest thanks of gratitude to my guide ***Prof. Prasun Das, SQC & OR Unit, Indian Statistical Institute, Kolkata,*** who gave me the opportunity to do this project under his guidance. This project gave me the scope to gain knowledge about the Supply Chain domain and it's practical implementation in FMCG domain.

I would also like to express my best regards and heartiest thanks of gratitude to ***my seniors and colleagues***, who guided me and helped me in completing the project. I have tried my best to learn as much as possible from them. I have consulted with the resources they referred to and many others. Their guidance and valuable inputs have enriched me and led to a wider view.

Belal Ahmed

Roll No.QR2304

M. Tech in QR & OR(2nd Year)

Indian Statistical Institute

Kolkata

TABLE OF CONTENTS

Abstract

This project explores the application of machine learning techniques to predict product weight (in tons) to be shipped for Fast-Moving Consumer Goods (FMCG). Utilizing a comprehensive dataset of warehouse attributes and performance metrics, we implement various regression models including Linear Regression, XGBoost, Decision Tree, and Random Forest to achieve high prediction accuracy. The models are evaluated based on R-squared scores and mean absolute error, with the XGBoost model demonstrating superior performance. The project emphasizes data preprocessing, feature engineering, and the use of advanced regression techniques to optimize predictive accuracy.

1.0 ***INTRODUCTION***

What is FMCG supply chain?

The Fast-Moving Consumer Goods (FMCG) supply chain refers to the process through which products in the FMCG category are manufactured, distributed, and delivered to consumers. FMCG products include items that are sold quickly and at relatively low cost, such as food and beverages, toiletries, over-the-counter drugs, and other consumables. The supply chain for these goods is complex and involves multiple stages and stakeholders, all aimed at ensuring products are available to consumers in a timely and cost-effective manner.

Key Characteristics of the FMCG Supply Chain

High Volume and Low Margins: The FMCG supply chain deals with large volumes of goods that are sold at low margins. Efficiency is critical to profitability.

Short Product Life Cycles: Many FMCG products have short shelf lives, necessitating fast and efficient supply chain operations.

Consumer-Centric: The supply chain is highly responsive to consumer demands and market trends.

Complex Logistics: The distribution network is extensive and complex, requiring precise coordination and management.

Technology Integration: Advanced technologies such as automation, data analytics, and supply chain management software are integral to modern FMCG supply chains.

The FMCG supply chain is a highly dynamic and intricate system designed to ensure the efficient production, distribution, and delivery of consumer goods. The success of an FMCG company largely depends on the effectiveness of its supply chain management, which directly impacts product availability, cost efficiency, and customer satisfaction.

2.0 ***Objective***

In this project, we tackled the challenge of predicting product weights in tons for a Fast-Moving Consumer Goods (FMCG) dataset. Our project involved several key steps:

1. **Visual Exploration**: We started by visualizing the data to get a clearer picture of how different features are distributed and related. This included creating count plots and box plots to explore how product weights varied across different categories.
2. **Feature Engineering and Preprocessing**: To make our models work better, we transformed categorical variables using one-hot encoding and analyzed these transformed features to improve model performance.
3. **Model Building and Evaluation**: We tested out several regression models, such as Linear Regression, XGBoost, Decision Tree Regressor, and Random Forest Regressor. By comparing their performance using metrics like R^2 scores and Mean Absolute Error, we found that the XGBoost was the most accurate, with an impressive R^2 score of about 0.9935
4. **Key Insights**: This analysis not only showcased the importance of choosing the right model but also provided insights into which features are most influential. The results offer practical guidance for enhancing operations and making strategic decisions in the FMCG sector

3.0 *Literature Review*

The application of machine learning (ML) in the FMCG supply chain has gained significant attention due to its potential to optimize operations, enhance decision-making, and improve overall efficiency. This literature review dives into various aspects of ML applications in the FMCG supply chain, highlighting key studies and their contributions.

- **Demand Forecasting**:

  - **Traditional Methods vs. ML**: Traditional demand forecasting methods often rely on historical sales data and basic statistical models. However, ML techniques, such as neural networks, support vector machines, and ensemble methods, have been shown to provide more accurate and robust forecasts. For instance, Choi et al. (2018) demonstrated that ML models could outperform traditional methods in predicting demand for FMCG products by capturing complex, nonlinear patterns in the data.
  - **Case Studies**: Numerous case studies have shown the effectiveness of ML in demand forecasting. A study by Ryu et al. (2020) on a large FMCG retailer revealed that using ML algorithms reduced forecast errors by 15%, leading to better inventory management and reduced stockouts.

- **Inventory Management**:

  - **Optimization Algorithms**: ML algorithms, particularly reinforcement learning, have been applied to optimize inventory levels. These algorithms can learn from the environment and adjust inventory policies dynamically. The work by Zhou et al. (2019) highlights the use of Q-learning in managing FMCG inventories, which resulted in a 10% reduction in holding costs.
  - **Predictive Analytics**: Predictive analytics powered by ML can foresee inventory issues such as overstock and stockouts. The integration of predictive models with real-time data analytics, as discussed by Gupta et al. (2018), has enabled FMCG companies to maintain optimal inventory levels, thus improving service levels and reducing costs

4.0 ***Data Description***

This dataset provides detailed information on various attributes of warehouses storing Fast-Moving Consumer Goods (FMCG). The dataset includes 21 features that capture different aspects of warehouse operations, management, and logistics. Each record in the dataset represents a specific warehouse identified by its unique ID, and the features encompass a wide range of information from physical attributes like storage capacity and location type to operational details such as refilling frequency, transport issues, and the number of retail shops served.

- **Ware_house_ID:** Product warehouse ID
- **WH_Manager_ID:** Employee ID of warehouse manager
- **Location_type:** Location of warehouse (city or village)
- **WH_capacity_size:** Storage capacity size of the warehouse
- **zone:** Zone of the warehouse
- **WH_regional_zone:** Regional zone of the warehouse under each zone
- **num_refill_req_l3m:** Number of times refilling has been done in last 3 months
- **transport_issue_l1y:** Any transport issue like accident or goods stolen reported in last one year
- **Competitor_in_mkt:** Number of instant noodles competitor in the market
- **retail_shop_num:** Number of retail shops selling the product under the warehouse area
- **wh_owner_type:** Company owning the warehouse or renting it
- **distributor_num:** Number of distributors working between the warehouse and retail shops
- **flood_impacted:** Indicator if the warehouse is in a flood-impacted area
- **flood_proof:** Indicator if the warehouse is flood-proof (e.g., storage at some height)
- **electric_supply:** Warehouse has electric backup like a generator for load shedding
- **dist_from_hub:** Distance between the warehouse and the production hub in kilometers
- **workers_num:** Number of workers in the warehouse
- **wh_est_year:** Warehouse establishment year
- **storage_issue_reported_l3m:** Storage issues reported to corporate office in last 3 months (e.g., rats, fungus due to moisture)
- **temp_reg_mach:** Indicator if the warehouse has a temperature regulating machine

- **approved_wh_govt_certificate:** Type of standard certificate issued to the warehouse from government regulatory body
- **wh_breakdown_l3m:** Number of times the warehouse faced a breakdown in last 3 months (e.g., strike, flood, electrical failure)
- **govt_check_l3m:** Number of times government officers visited the warehouse to check quality and expiry of stored food in last 3 months
- **product_wg_ton:** Product shipped in last 3 months (weight in tons)

5.0 ***Methodology***

The methodology of this project involves these steps.

1. Data Collection:

- **Source:** The dataset used for the project was collected from a warehouse management system.
- **Content:** The dataset consists of 25,000 rows and 21 columns, which include various features related to warehouse characteristics and utilization.

2. Data Preprocessing:

- **Loading Data:** The dataset was loaded into a pandas DataFrame.
- **Missing Values:** Checked for missing values in the dataset using df.isna().sum(). Appropriate handling was applied if any missing values were found.
- **Descriptive Statistics:** Used df.describe() to get an overview of the data distribution for each feature.

3. Exploratory Data Analysis (EDA):

- **Histograms:** Plotted histograms of numerical features to understand their distributions.
- **Boxplots:** Used seaborn's boxplot to visualize the distribution and outliers of numerical features.
- **Count Plots:** Created count plots for categorical features, segmented by the zone feature to understand the distribution of categorical variables.

4. Feature Engineering:

- **Correlation Analysis:** Generated a heatmap of feature correlations using seaborn's heatmap to identify multicollinearity and relationships between features.
- **One-Hot Encoding:** Applied one-hot encoding to categorical variables (zone and WH_regional_zone) using ColumnTransformer and OneHotEncoder from scikit-learn.

5. Model Training and Evaluation:

- **Data Splitting:** Split the dataset into training and testing sets using train_test_split with a test size of 20%.

- **Linear Regression:**
  - **Model Training:** Trained a Linear Regression model using the training data.
  - **Evaluation:** Calculated the R^2 score on the test data.
- **XGBoost Regressor:**
  - **Model Training:** Trained an XGBoost Regressor using the training data.
  - **Evaluation:** Calculated the R^2 score on the test data.
- **Decision Tree Regressor:**
  - **Model Training:** Trained a Decision Tree Regressor using the training data.
  - **Evaluation:** Calculated the R^2 score on the test data.
- **Random Forest Regressor:**
  - **Model Training:** Trained a Random Forest Regressor with 500 estimators using the training data.
  - **Evaluation:** Calculated the R^2 score on the test data.

6. Model Performance Comparison:

- **Linear Regression:** Achieved an R^2 score of 0.976 on the test data.
- **XGBoost Regressor:** Achieved an R^2 score of 0.9935 on the test data.
- **Decision Tree Regressor:** Achieved an R^2 score of 0.987 on the test data.
- **Random Forest Regressor:** Achieved an R^2 score of 0.9920 on the test data.

7. Error Analysis:

- **Mean Absolute Error (MAE):** Calculated the MAE for the XGBoost Regressor, which was 700.76.

5.1. ***Data Collection***

- **Source:**
  - The data was sourced from a warehouse management system. This system tracks various operational metrics and attributes related to warehouse performance and capacity.
- **Content:**
  - The dataset includes 25,000 records (rows) and 21 attributes (columns). Each record represents a specific warehouse , detailing its operational characteristics and the **target variable**, product_wg_ton, which indicates the product weight (in tons) shipped in the last 3 months. This variable is crucial as it directly measures the capacity utilization of the warehouses.

5.2. *Data Preprocessing*

Loading Data:

- The dataset was loaded into a pandas DataFrame, a powerful data structure from the pandas library that allows for efficient data manipulation and analysis.

Handling Missing Values:

- The presence of missing values was assessed. Missing values can skew the analysis and modeling if not handled appropriately. Various strategies, such as imputation with mean and mode, or removing rows/columns with missing values, were considered based on the nature and extent of the missing data.

Descriptive Statistics:

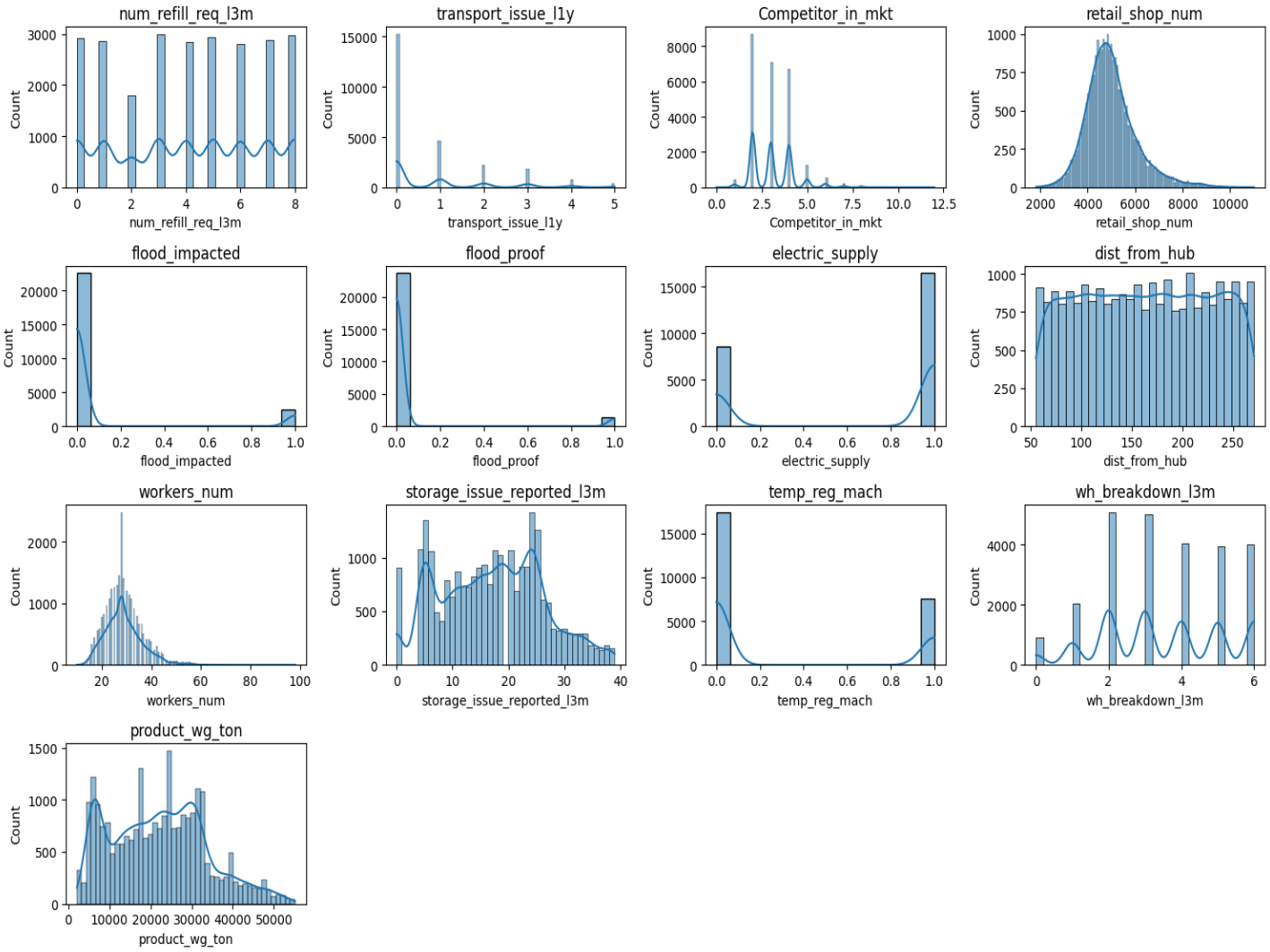
- Generated a statistical summary to understand the central tendency (mean, median), dispersion (standard deviation, interquartile range), and distribution (min, max, percentiles) of the data. This step helps in identifying outliers and understanding the overall data distribution.

5.3. Exploratory Data Analysis.

In EDA,several plots have been created to get better idea of the dataset.

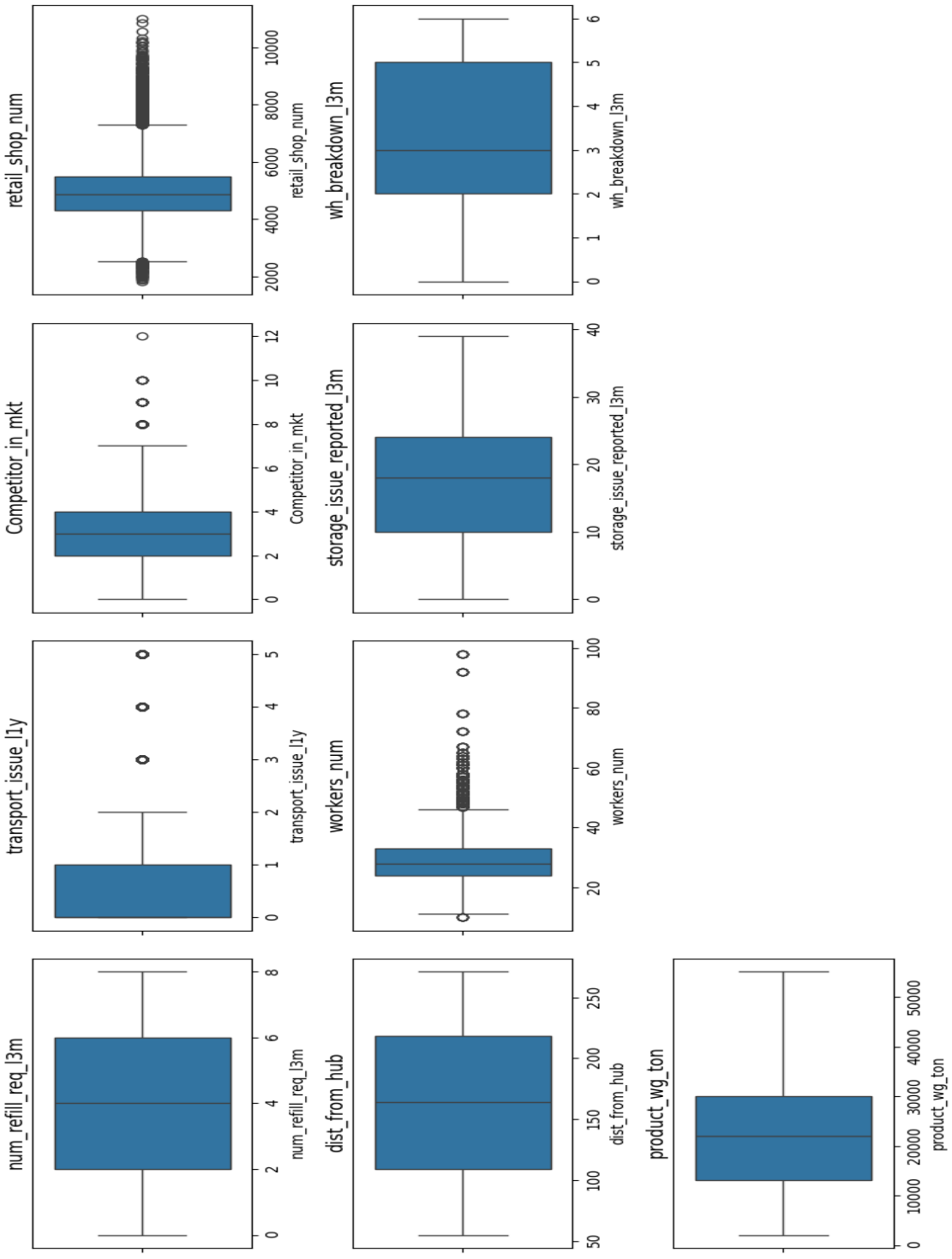
Histograms:

- **Purpose:**
  - To understand the distribution of numerical features.
- **Process:**
  - Created histograms for each numerical feature in the dataset. Histograms plot the frequency of data points within specified intervals (bins).



Boxplots:

- **Purpose:**
  - To visualize the spread and identify outliers in numerical features.
- **Process:**
  - Boxplots for each numerical feature were plotted. Boxplots display the median, quartiles, and potential outliers in the data.



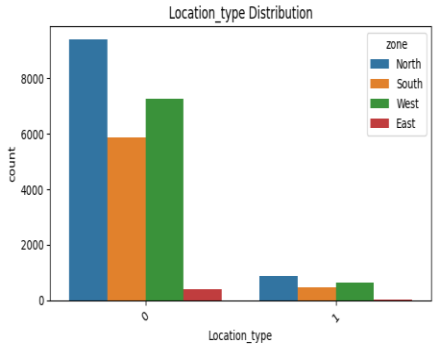
Count Plots for Categorical Features:

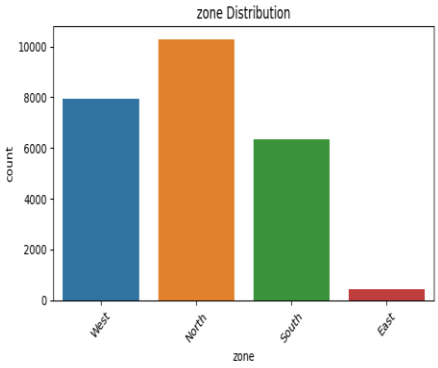
Purpose:

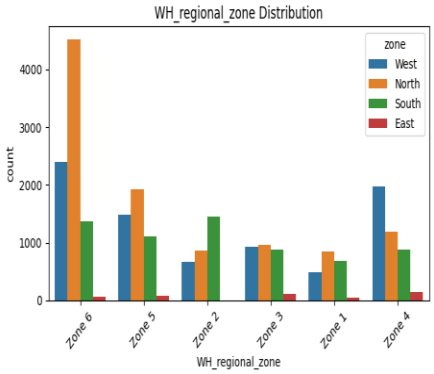
- To observe the distribution of categorical features within the dataset.
- To identify the frequency of different categories and potential imbalances.
- To provide insights into how categorical features might influence the target variable ‘product_wg_ton’.

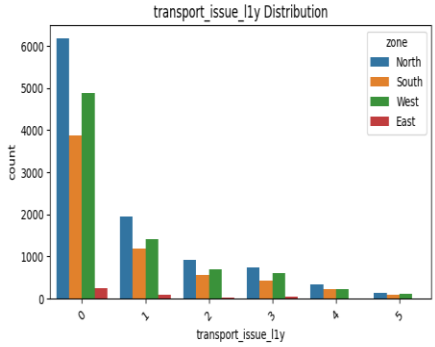
Process:

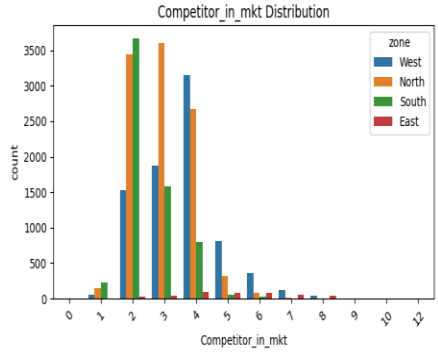
- **Selection of Categorical Features:**
  - Identified key categorical features including Location_type, zone, WH_regional_zone, transport_issue_l1y (transport issues in the last year), Competitor_in_mkt (presence of competitors in the market), wh_owner_type (type of warehouse ownership), flood_impacted (whether the warehouse was impacted by floods), flood_proof (whether the warehouse is flood-proof), electric_supply (availability of electric supply), storage_issue_reported_l3m (storage issues reported in the last 3 months), temp_reg_mach (availability of temperature regulation machines), approved_wh_govt_certificate (whether the warehouse has government approval), wh_breakdown_l3m (warehouse breakdowns in the last 3 months), and govt_check_l3m (government checks in the last 3 months).
- **Plot Generation:**
  - For each categorical feature, created count plots to visualize the number of occurrences of each category.
  - Used the ‘zone’ feature as a hue to provide additional segmentation and to observe the distribution across different zones.

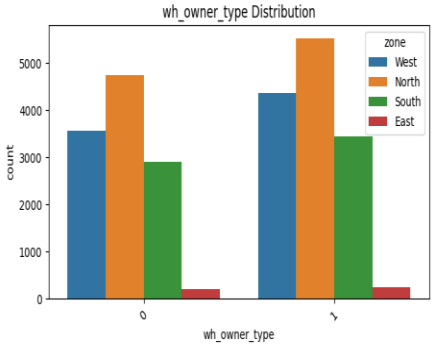

Location_type Distribution
zone
North
South
West
East
count
Location_type

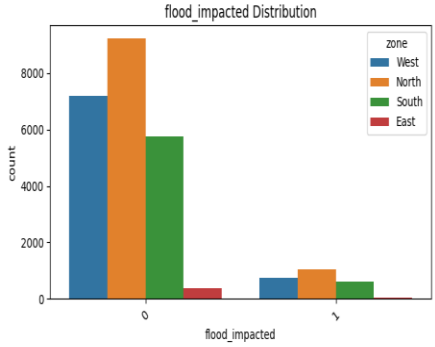

zone Distribution
count
West
North
South
East
zone

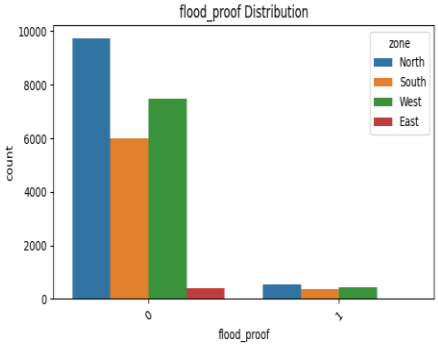

WH_regional_zone Distribution
zone
West
North
South
East
count
Zone 6
Zone 5
Zone 2
Zone 3
Zone 1
Zone 4
WH_regional_zone

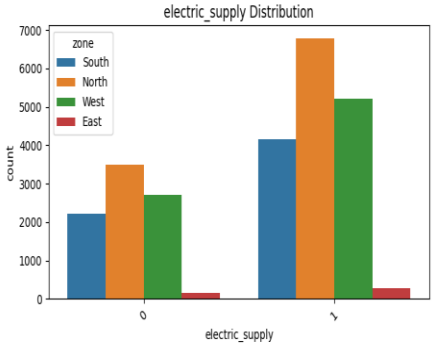

transport_issue_l1y Distribution
zone
North
South
West
East
count
transport_issue_l1y

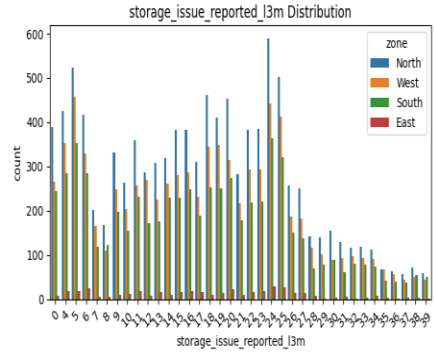

Competitor_in_mkt Distribution
zone
West
North
South
East
count
Competitor_in_mkt

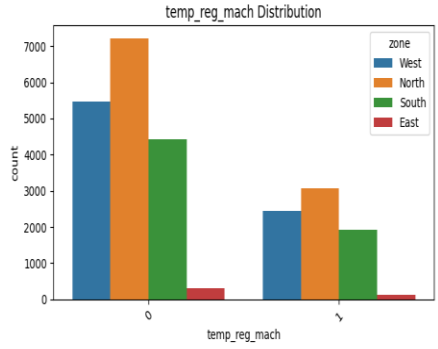

wh_owner_type Distribution
zone
West
North
South
East
count
wh_owner_type

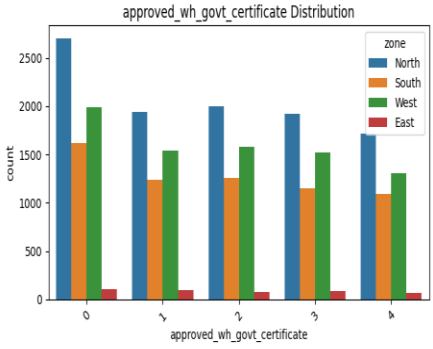

flood_impacted Distribution
zone
West
North
South
East
count
flood_impacted

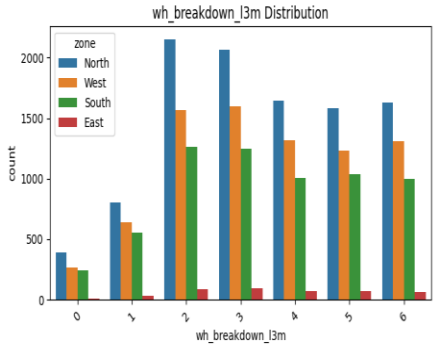

flood_proof Distribution
zone
North
South
West
East
count
flood_proof

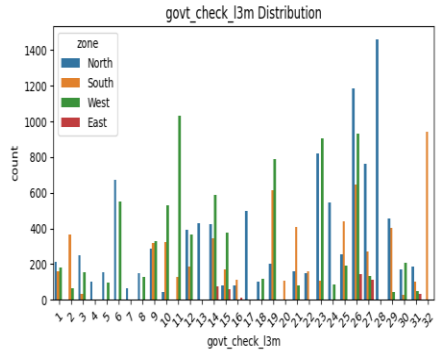

electric_supply Distribution
zone
South
North
West
East
count
electric_supply


storage_issue_reported_l3m Distribution
zone
North
West
South
East
count
storage_issue_reported_l3m


temp_reg_mach Distribution
zone
West
North
South
East
count
temp_reg_mach


approved_wh_govt_certificate Distribution
zone
North
South
West
East
count
approved_wh_govt_certificate


wh_breakdown_l3m Distribution
zone
North
West
South
East
count
wh_breakdown_l3m


govt_check_l3m Distribution
zone
North
South
West
East
count
govt_check_l3m

5.4 Feature Engineering.

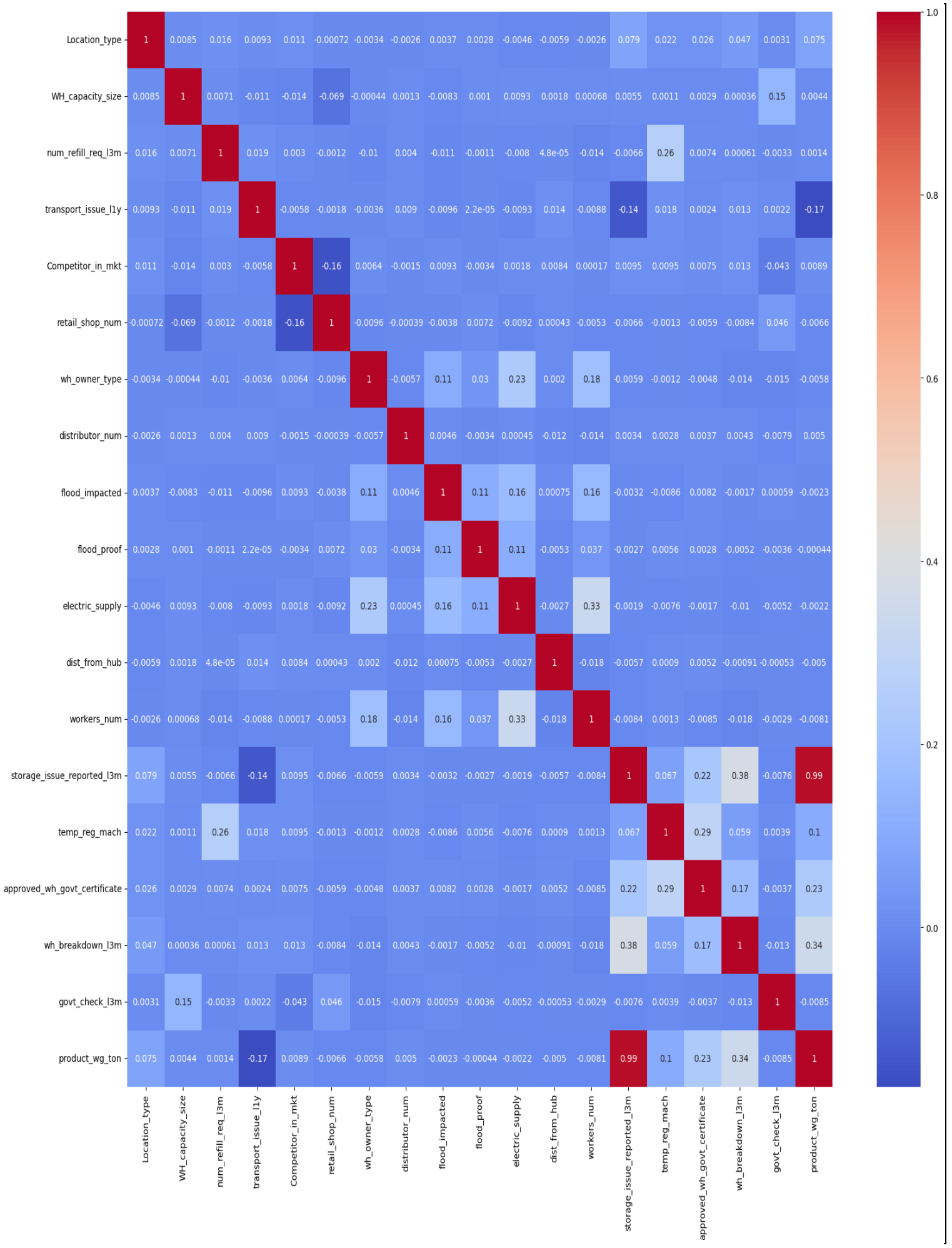
Heatmap of Correlation Matrix:

Purpose:

- To visualize the relationships between numerical features in the dataset.
- To identify which features are strongly correlated with each other, either positively or negatively.
- To detect potential multicollinearity among features, which can influence the effectiveness and stability of predictive models.

Process:

- **Correlation Matrix Computation:**
  - Calculated the correlation matrix for all numerical features in the dataset. This matrix shows the pairwise correlation coefficients between features, ranging from -1 to 1.
  - A correlation coefficient close to 1 indicates a strong positive correlation, meaning as one feature increases, the other feature also increases.
  - A correlation coefficient close to -1 indicates a strong negative correlation, meaning as one feature increases, the other feature decreases.
  - A correlation coefficient around 0 indicates no linear relationship between the features.


Location_type 1 0.0085 0.016 0.0093 0.011 -0.00072 -0.0034 -0.0026 0.0037 0.0028 -0.0046 -0.0059 -0.0026 0.079 0.022 0.026 0.047 0.0031 0.075
WH_capacity_size 0.0085 1 0.0071 -0.011 -0.014 -0.069 -0.00044 0.0013 -0.0083 0.001 0.0093 0.0018 0.00068 0.0055 0.0011 0.0029 0.00036 0.15 0.0044
num_refill_req_l3m 0.016 0.0071 1 0.019 0.003 -0.0012 -0.01 0.004 -0.011 -0.0011 -0.008 4.8e-05 -0.014 -0.0066 0.26 0.0074 0.00061 -0.0033 0.0014
transport_issue_l1y 0.0093 -0.011 0.019 1 -0.0058 -0.0018 -0.0036 0.009 -0.0096 2.2e-05 -0.0093 0.014 -0.0088 -0.14 0.018 0.0024 0.013 0.0022 -0.17
Competitor_in_mkt 0.011 -0.014 0.003 -0.0058 1 -0.16 0.0064 -0.0015 0.0093 -0.0034 0.0018 0.0084 0.00017 0.0095 0.0095 0.0075 0.013 -0.043 0.0089
retail_shop_num -0.00072 -0.069 -0.0012 -0.0018 -0.16 1 -0.0096 -0.00039 -0.0038 0.0072 -0.0092 0.00043 -0.0053 -0.0066 -0.0013 -0.0059 -0.0084 0.046 -0.0066
wh_owner_type -0.0034 -0.00044 -0.01 -0.0036 0.0064 -0.0096 1 -0.0057 0.11 0.03 0.23 0.002 0.18 -0.0059 -0.0012 -0.0048 -0.014 -0.015 -0.0058
distributor_num -0.0026 0.0013 0.004 0.009 -0.0015 -0.00039 -0.0057 1 0.0046 -0.0034 0.00045 -0.012 -0.014 0.0034 0.0028 0.0037 0.0043 -0.0079 0.005
flood_impacted 0.0037 -0.0083 -0.011 -0.0096 0.0093 -0.0038 0.11 0.0046 1 0.11 0.16 0.00075 0.16 -0.0032 -0.0086 0.0082 -0.0017 0.00059 -0.0023
flood_proof 0.0028 0.001 -0.0011 2.2e-05 -0.0034 0.0072 0.03 -0.0034 0.11 1 0.11 -0.0053 0.037 -0.0027 0.0056 0.0028 -0.0052 -0.0036 -0.00044
electric_supply -0.0046 0.0093 -0.008 -0.0093 0.0018 -0.0092 0.23 0.00045 0.16 0.11 1 -0.0027 0.33 -0.0019 -0.0076 -0.0017 -0.01 -0.0052 -0.0022
dist_from_hub -0.0059 0.0018 4.8e-05 0.014 0.0084 0.00043 0.002 -0.012 0.00075 -0.0053 -0.0027 1 -0.018 -0.0057 0.0009 0.0052 -0.00091 -0.00053 -0.005
workers_num -0.0026 0.00068 -0.014 -0.0088 0.00017 -0.0053 0.18 -0.014 0.16 0.037 0.33 -0.018 1 -0.0084 0.0013 -0.0085 -0.018 -0.0029 -0.0081
storage_issue_reported_l3m 0.079 0.0055 -0.0066 -0.14 0.0095 -0.0066 -0.0059 0.0034 -0.0032 -0.0027 -0.0019 -0.0057 -0.0084 1 0.067 0.22 0.38 -0.0076 0.99
temp_reg_mach 0.022 0.0011 0.26 0.018 0.0095 -0.0013 -0.0012 0.0028 -0.0086 0.0056 -0.0076 0.0009 0.0013 0.067 1 0.29 0.059 0.0039 0.1
approved_wh_govt_certificate 0.026 0.0029 0.0074 0.0024 0.0075 -0.0059 -0.0048 0.0037 0.0082 0.0028 -0.0017 0.0052 -0.0085 0.22 0.29 1 0.17 -0.0037 0.23
wh_breakdown_l3m 0.047 0.00036 0.00061 0.013 0.013 -0.0084 -0.014 0.0043 -0.0017 -0.0052 -0.01 -0.00091 -0.018 0.38 0.059 0.17 1 -0.013 0.34
govt_check_l3m 0.0031 0.15 -0.0033 0.0022 -0.043 0.046 -0.015 -0.0079 0.00059 -0.0036 -0.0052 -0.00053 -0.0029 -0.0076 0.0039 -0.0037 -0.013 1 -0.0085
product_wg_ton 0.075 0.0044 0.0014 -0.17 0.0089 -0.0066 -0.0058 0.005 -0.0023 -0.00044 -0.0022 -0.005 -0.0081 0.99 0.1 0.23 0.34 -0.0085 1
Location_type
WH_capacity_size
num_refill_req_l3m
transport_issue_l1y
Competitor_in_mkt
retail_shop_num
wh_owner_type
distributor_num
flood_impacted
flood_proof
electric_supply
dist_from_hub
workers_num
storage_issue_reported_l3m
temp_reg_mach
approved_wh_govt_certificate
wh_breakdown_l3m
govt_check_l3m
product_wg_ton
1.0
0.8
0.6
0.4
0.2
0.0

One Hot Encoding

- Categorical variables, such as zone and WH_regional_zone, were converted into numerical format using one-hot encoding. This process creates binary columns for each category level, enabling the use of these categorical features in machine learning models without introducing ordinal relationships.

.

5.5 Model Training and Evaluation

- **Data Splitting:**
  - The dataset was split into training and testing sets. Typically, 80% of the data is used for training the model, and 20% is used for evaluating the model's performance. This split helps in assessing the model's generalizability to unseen data.
- **Linear Regression:**
  - A Linear Regression model was trained on the training data. This model assumes a linear relationship between the input features and the target variable, `product_wg_ton`.
  - The model's performance was evaluated using the R^2 score, which indicates the proportion of variance in the target variable explained by the input features.
- **XGBoost Regressor:**
  - An XGBoost Regressor, a powerful gradient boosting model, was trained. This model is capable of capturing complex, non-linear relationships between features and the target variable.
  - The model's performance was evaluated using the R^2 score, demonstrating its ability to predict `product_wg_ton` accurately.
- **Decision Tree Regressor:**
  - A Decision Tree Regressor was trained. This model splits the data into subsets based on feature values, creating a tree-like structure for prediction.
  - The model's performance was evaluated using the R^2 score, highlighting its ability to handle non-linear relationships.
- **Random Forest Regressor:**
  - A Random Forest Regressor, an ensemble of decision trees, was trained. This model aggregates the predictions from multiple trees to improve accuracy and reduce overfitting.
  - The model's performance was evaluated using the R^2 score, showcasing its robustness and accuracy in predicting `product_wg_ton`.

5.6 Model Performance and Comparison.

Comparison of R^2 Scores:

- The performance of the different models was compared based on their R^2 scores on the test data:
    - Linear Regression: Achieved an R^2 score of 0.976, indicating a strong linear relationship but with some room for improvement.
    - XGBoost Regressor: Achieved an R^2 score of 0.9935, demonstrating excellent performance with the ability to capture complex relationships.
    - Decision Tree Regressor: Achieved an R^2 score of 0.987, showing good performance but slightly less accurate than the XGBoost model.
    - Random Forest Regressor: Achieved an R^2 score of 0.9920, matching the XGBoost Regressor in accuracy, reflecting the effectiveness of ensemble methods.

5.7 Error Analysis.

Mean Absolute Error (MAE):

- The MAE was calculated for the XGBoost Regressor to measure the average magnitude of errors in the predictions. The MAE provides an intuitive understanding of the prediction errors in the same units as `product_wg_ton`

6 *Result and Discussion*

The performance of different models highlighted varying degrees of predictive accuracy. The **XGBoost Regressor** emerged as the top performer with an R2 score of 0.9935, which means it explained 99.35% of the variance in the target variable. This high performance can be attributed to XGBoost's ability to handle complex relationships and interactions within the data through its gradient boosting framework. The model also had the lowest mean absolute error, demonstrating superior accuracy in predicting product weights compared to the other models.

The **Random Forest Regressor** also performed very well, with an R2 score of 0.992. This model benefits from its ensemble approach, combining multiple decision trees to improve prediction accuracy and robustness. However, it showed slightly less precision than XGBoost, which may be due to the latter's more advanced boosting techniques.

The **Decision Tree Regressor** showed solid performance with an R2 score of 0.987. While effective, decision trees are prone to overfitting and may not generalize as well on unseen data compared to ensemble methods. This result highlights the trade-off between model complexity and interpretability.

The **Linear Regression** model provided a baseline with an R2 score of 0.976. While simpler and more interpretable, it fell short compared to the more complex models. This lower score suggests that linear regression might not capture the intricate patterns in the data as effectively as XGBoost or Random Forest.

Overall, the results indicate that advanced ensemble models like XGBoost and Random Forest are more effective for this regression task compared to simpler models, underscoring the benefit of using complex algorithms to handle intricate relationships in the dataset.

7 *Conclusion and Future Scope of Work*

The analysis demonstrates that advanced ensemble models, particularly the XGBoost Regressor and Random Forest Regressor, significantly outperform simpler models such as Linear Regression and Decision Tree Regressor in predicting the weight of products shipped. The XGBoost model achieved the highest predictive accuracy, explaining 99.35% of the variance in the target variable, which suggests its robustness in capturing complex patterns in the data. The Random Forest Regressor also performed exceptionally well, validating the effectiveness of ensemble methods in improving prediction accuracy and generalization. The results underscore the value of sophisticated algorithms in handling real-world data with multiple interacting features.

Future Scope of Work

1. **Feature Engineering**: Further refinement of feature engineering could enhance model performance. Exploring interaction terms, polynomial features, or domain-specific variables might uncover additional insights and improve predictive accuracy.
2. **Model Tuning and Optimization**: While the current models are effective, there is potential for further improvement through hyperparameter tuning. Implementing techniques like grid search or random search for parameter optimization could lead to better performance.
3. **Incorporation of External Data**: Integrating additional external data sources, such as market trends or weather conditions, could provide more context and improve the models' predictive capabilities.
4. **Advanced Models**: Exploring other advanced models, such as Neural Networks or Support Vector Machines, could offer further insights and potentially enhance prediction accuracy.
5. **Cross-Validation**: Implementing more robust cross-validation techniques could provide a better assessment of model performance and ensure its reliability across different subsets of the data.

References

Tarallo, E., Akabane, G. K., Shimabukuro, C. I., Mello, J., & Amancio, D.. *Machine Learning in Predicting Demand for Fast-Moving Consumer Goods: An Exploratory Research*.

Carbonneau, R. A., Vahidov, R., & Laframboise, K. (2007). *Machine Learning-Based Demand Forecasting in Supply Chains*. International Journal of Intelligent Information Technologies

Suwignjo, P., Panjaitan, L., Baihaqy, A. R., & Rusdiansyah, A. (2023). *Predictive Analytics to Improve Inventory Performance: A Case Study of an FMCG Company*. Operations and Supply Chain Management: An International Journal.

Mebal, A., Hema, S., Jothika, S. J., & Manochitra, M. (2021). *Predicting the Demand for FMCG using Machine Learning*. International Journal of Engineering and Advanced Technology (IJEAT).

Appendix

```
# Importing necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_absolute_error
from xgboost import XGBRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor

# Load the dataset
df = pd.read_csv("fmcg_data.csv")  # Replace with your actual dataset path

# Initial data exploration
print(df.head())  # Display the first few rows of the dataset
print(df.info())  # Display information about the dataset
print(df.describe())  # Display statistical summary of the dataset

# Data cleaning (if needed)
# For example, handling missing values
df.fillna(df.median(), inplace=True)

# Data visualization: Correlation heatmap for numeric features
plt.figure(figsize=(20, 20))
sns.heatmap(df.corr(numeric_only=True), cmap="coolwarm", annot=True)
plt.title("Correlation Heatmap of Numeric Features")
plt.show()

# Data visualization: Distribution of the target variable
plt.figure(figsize=(10, 6))
sns.histplot(df['product_wg_ton'], bins=30, kde=True)
plt.title('Distribution of product_wg_ton')
plt.xlabel('Product Weight in Tons')
plt.ylabel('Frequency')
plt.show()

# Data visualization: Boxplots for selected numeric features
numeric_features = ['num_refill_req_l3m', 'retail_shop_num', 'distributor_num', 'dist_from_hub',
'workers_num']
plt.figure(figsize=(20, 20))
```

```
for i, feature in enumerate(numeric_features, 1):
    plt.subplot(3, 2, i)
    sns.boxplot(data=df, x='WH_capacity_size', y=feature)
    plt.title(f'{feature} by WH_capacity_size')
plt.tight_layout()
plt.show()

# Data visualization: Count plots for categorical features
categorical_features = [
    'Location_type', 'zone', 'WH_regional_zone', 'transport_issue_l1y', 'Competitor_in_mkt',
'wh_owner_type',
    'flood_impacted', 'flood_proof', 'electric_supply', 'storage_issue_reported_l3m', 'temp_reg_mach',
    'approved_wh_govt_certificate', 'wh_breakdown_l3m', 'govt_check_l3m'
]
plt.figure(figsize=(20, 20))
for i, feature in enumerate(categorical_features, 1):
    plt.subplot(5, 3, i)
    sns.countplot(data=df, x=feature, hue="zone")
    plt.xticks(rotation=45)
    plt.title(f'{feature} Distribution')
plt.tight_layout()
plt.show()

# Splitting data into features and target variable
X = df.iloc[:, :-1]
y = df.iloc[:, -1]

# Splitting data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Preprocessing: One-Hot Encoding for categorical variables
transformer = ColumnTransformer(
    transformers=[
        ("OHE", OneHotEncoder(sparse_output=False, drop="first"), ["zone", "WH_regional_zone"])
    ], remainder="passthrough"
)

# Applying the transformations
X_train = transformer.fit_transform(X_train)
X_test = transformer.transform(X_test)

# Model training and evaluation: Linear Regression
lr = LinearRegression()
lr.fit(X_train, y_train)
y_pred_lr = lr.predict(X_test)
print(f'Linear Regression R2 Score: {r2_score(y_test, y_pred_lr)}')

# Model training and evaluation: XGBoost Regressor
```

```
xgb = XGBRegressor()
xgb.fit(X_train, y_train)
print(f'XGBoost Regressor R2 Score: {xgb.score(X_test, y_test)}')

# Model training and evaluation: Decision Tree Regressor
dtr = DecisionTreeRegressor()
dtr.fit(X_train, y_train)
print(f'Decision Tree Regressor R2 Score: {dtr.score(X_test, y_test)}')

# Model training and evaluation: Random Forest Regressor
rfr = RandomForestRegressor(n_estimators=500, n_jobs=-1)
rfr.fit(X_train, y_train)
print(f'Random Forest Regressor R2 Score: {rfr.score(X_test, y_test)}')

# Evaluating the best model (XGBoost) on the whole dataset
X_transformed = transformer.fit_transform(X)
print(f'XGBoost Regressor R2 Score on Full Data: {xgb.score(X_transformed, y)}')
```