CAPSTONE PROJECT FINAL REPORT ON

SUPPLY CHAIN

PROJECT

DSBA

BY: SUBHAM AGRAWAL &
ARPIT DONERIA

Table of Contents

Table of Figures

1. Executive Summary:

Problem Statement :

A FMCG company has introduced a new poduct in the market which is noodels and facing various issues regarding its inventory management, like in many areas there are less demand for the product but the company has delivered more Similarly where the demand is more the company is sending less supplies because of not estimating the proper demand.

Data Description:

The given data is the details of the accounts and their various attributes related to the Warehouses of the company and its issues. It consists of 25000 rows and 24 columns where rows depict the number of warehouses and the columns depict the various attributes. Out of the 24 attributes, 8 attributes are of object data type and remaining are either integer or float data types. However, if there are null values or invalid data entry, this will be imputed, and the data type conversions may be required as we proceed further.

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

Main Results:

Models	Adjusted R^2		Root mean squared error (RMSE)		Rank of models
	Training	Testing	Training	Testing	
Linear Regression	99.03	99	1153.848	1118.162	4
Random forest Regressor	99.48	99.37	838.217	908.598	2
Neural Networks	99.11	99.01	1090.642	1149.596	3
K-Nearest Neighbors Regressor	75.7	48.8	32681879	8305	5
Gradient Boosting	99.55	99.31	781.213	961.711	1

- In this we found out That Gradient Boosting is the best model which has the lowest Mean square error in training and also highest accuracy score.
- We can also take Random forest model into consideration since in test data it is maximum close to the training data and also the lowest but the reason why it is been given 2nd rank because it shows more mean square error in the training data.
- The K-nearest neighbor is the worst model and cannot be considered for model building of the data. It shows an overfitting in the models and also highest RMSE in the data.
- We also see that neural networks and linear regression model is also a good model with a very high accuracy score of 99% but they are not the best model in here because other model shows a higher accuracy rate and lower RMSE.

Recommendations:

- The company needs to manage its unnecessary cost which it is giving like for theft of goods, storage issues in the warehouses, number of workers working in a particular warehouse, flood impacted zones need to have special care and appropriate measures should be taken so that the product doesn't get destroyed.
- In the Flood impacted zones and flood proof zones also there is a big number difference the company should take appropriate steps to make the warehouses flood proofs at least in the flood impacted zones Also we see there is a significant difference in the sale of warehouses having temperature regulating machines which are helping the goods to sustain for longer times.
- The company needs to fulfill the demand according to the data present like we see there are few number of stores in the rural areas but there is a huge demand of noodles from that area where as people from rural area are not consuming noodels upto that extend so company needs to focus on rural areas much strongly.
- Government Certification plays a vital role in the sale of products of the business therefore we recommend to increase the rankings in the certificates as we see better the rank better is the sale for that particular warehouse

These are some of the Recoomendations and insights for more please go through whole project file.

Introduction:

In this business problem we have received the data of a FMCG company which is trying to control its inventory management cost in different zones of the country.

The company has launched a new product in the market which is noodles and facing various issues regarding its inventory management, like in many areas there are less demand for the product but the company has delivered more Similarly where the demand is more the company is sending less supplies because of not estimating the proper demand.

A supply chain management plays a very vital role in conducting the balance and smooth running of the business and if the company does not identify its potential market then the company won't be able to grow with its full potential.

A simple solution to this problem can be identifying the demand and supply of the product according to the areas they are delivering in. This can be done by looking at the past data and various aspects of where the company is unable to fulfil the demands, other than demand and supply what all are the factors due to which the company is unable to fulfil the inventory demands like flood impacts, loss in transport, Theft issues, electricity issues, proper marketing or brand value of the product, understanding the competitors' products what good or bad things they are doing, understanding the nature of their suppliers etc. which can be discussed.

Objectives:

The reason why we are doing this project are as follows:-

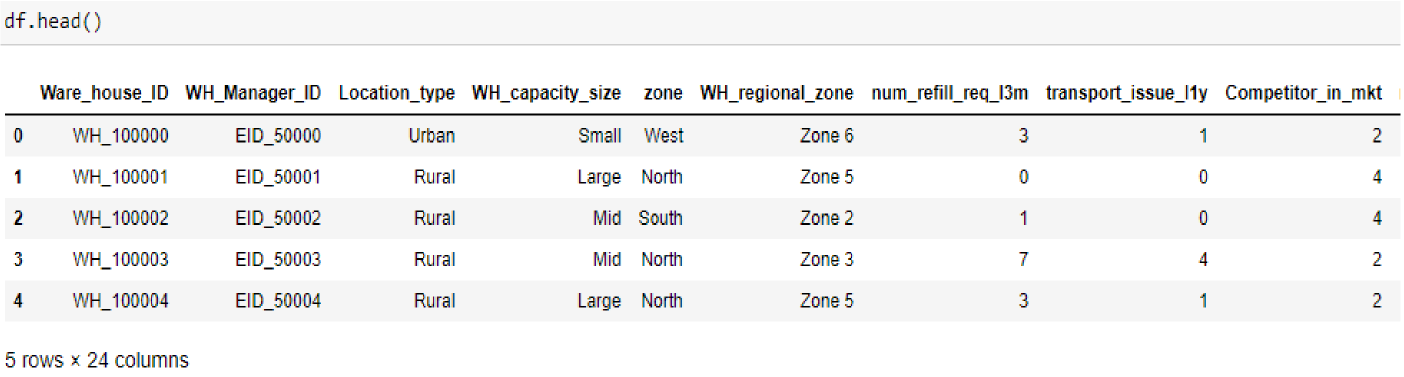
- To predict the quantity of the products of a FMCG company for different warehouses and their respective zones.
- To reduce wastage of the product which result in loss to the company
- To identify its potential market according to its geographical locations
- To build a model for Various FMCG companies in order to make their supply chain working better
- To suggest various changes it needs to be done in its company policies in order to make their transit better.
- Demand forecasting for their products
- Identifying the importance of supply chain for effective running of the company
- This model can also help to launch new and similar products once the current issues are resolved which will help in growing the overall business
- This project will help the stake holders identifying their potential market and making its root more stronger by reaching out to maximum people over their
- Once the stake holders identify the areas that are non-performing they can even spend more on marketing in order to reach more number of people and spreading more awareness about the product.

In this project we have built various regression models like linear regression, K-Nearest Neighbors Regression, Random forest Regressor, Gradient Boosting regressor, Neural Networks to identify which model suits best for the given data. And then we will predict the outcomes according to the model we will select and build and see how much accurate is the data to test it to the real world.

Constraints: There is no clear information about the production units and distance of the warehouses from their, Also no information provided about the sale in retail stores by the company, also the regional zones are not easily understood of which regional zone belongs to which zone.

2. EDA and its Insights

This is a huge data with 25000 entries and 24 columns inside it.

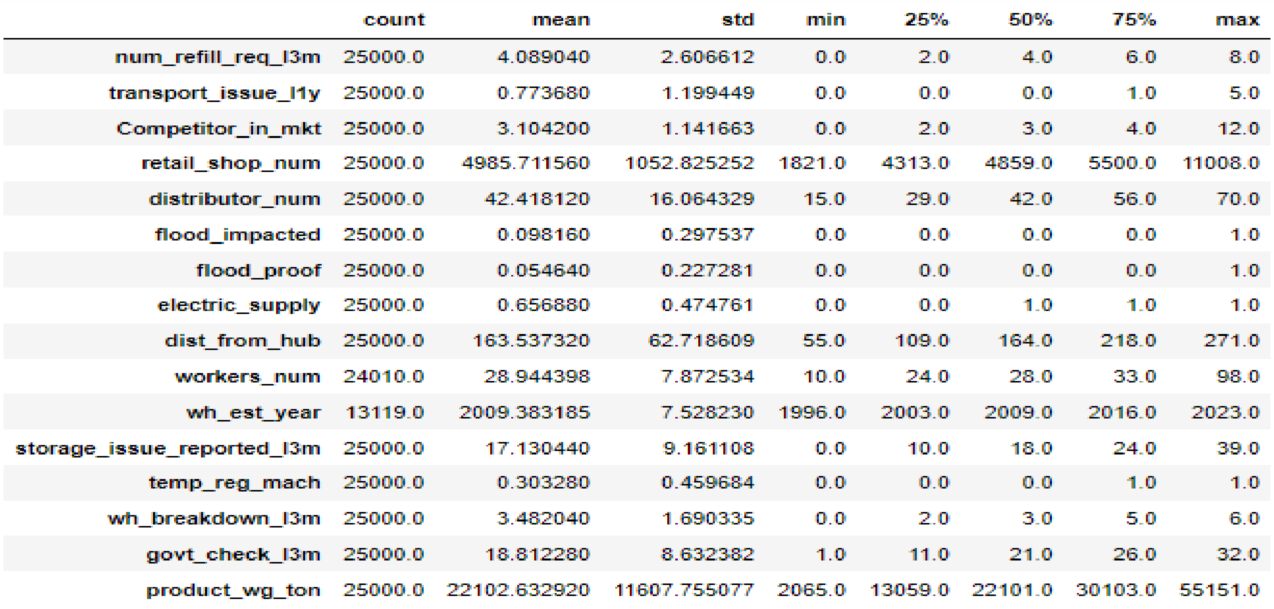


```
df.head()
```

	Ware_house_ID	WH_Manager_ID	Location_type	WH_capacity_size	zone	WH_regional_zone	num_refill_req_l3m	transport_issue_l1y	Competitor_in_mkt
0	WH_100000	EID_50000	Urban	Small	West	Zone 6	3	1	2
1	WH_100001	EID_50001	Rural	Large	North	Zone 5	0	0	4
2	WH_100002	EID_50002	Rural	Mid	South	Zone 2	1	0	4
3	WH_100003	EID_50003	Rural	Mid	North	Zone 3	7	4	2
4	WH_100004	EID_50004	Rural	Large	North	Zone 5	3	1	2

5 rows × 24 columns

In this data we do have missing values in three columns in which is been treated and 1st column which is worker num is been replaced by its median value 2nd column which is approved govt certificates in which missing values are being replaced by unknown rows and 3rd column which is wh_est_year is been removed since the missing values were more than 40%.



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

In the describe function of the data it shows data is mostly normally distributed as most of means and medians are almost equal in the data set and there is not much of a skewness in here.

There are no duplicate Rows in the dataset.

There are 6 Categorical and 18 numerical columns currently present which we need to treat and remove some unwanted columns from the data. Also we have to convert some numerical columns to categorical ones in order to perform a better EDA at the data

Firstly we have to remove the unwanted columns present in our data because it may mislead our EDA analysis

So in this data we have removed ware house ID and WH manager ID because these columns won't help us anyway in interpreting any kind of results of supply chain

Univariate Analysis

In Univariate Analysis we understand the distribution of each and every column for various numeric and categorical Columns.
For numeric Columns we have seen various Displot and boxplot for identifying the outliers in the numeric columns or see the distribution of various columns through displot. For Categorical columns we are seeing countplot for each variable and identifying the distributions for it here we identify how much weightage is been given to each and every variable in the column and how to interpret it for further analysis.

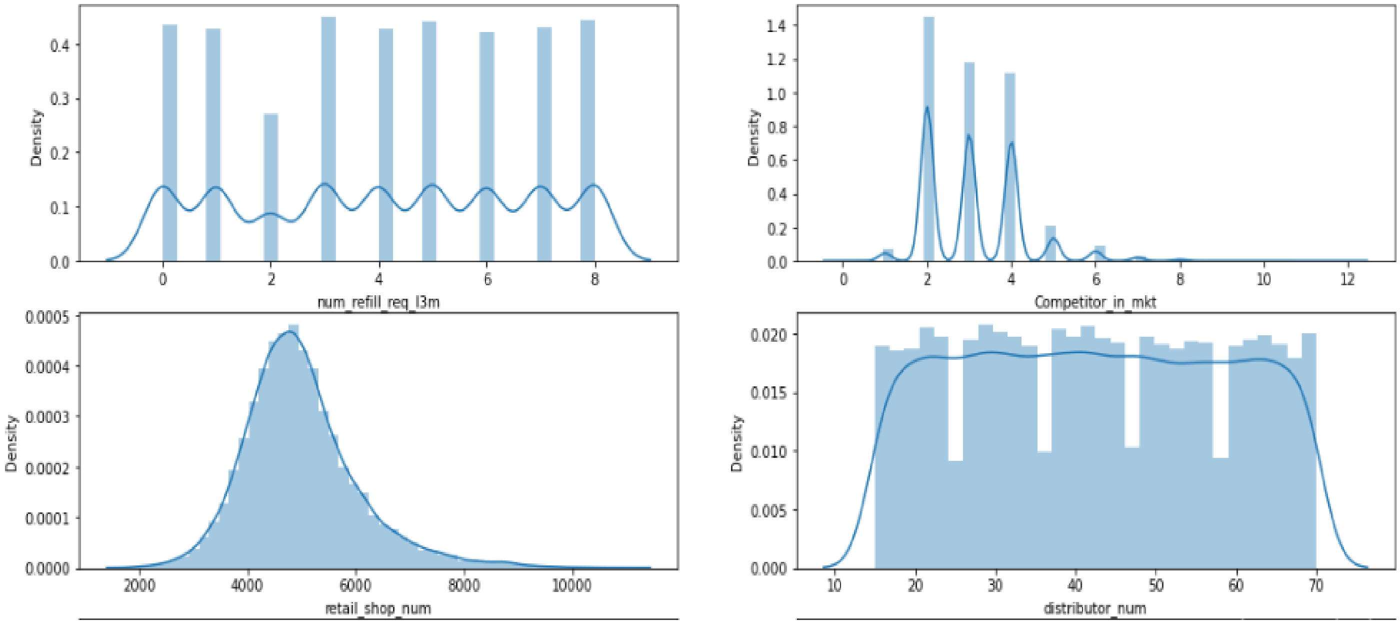


Figure 1 –Graphs of competitors, retail shops and distributors

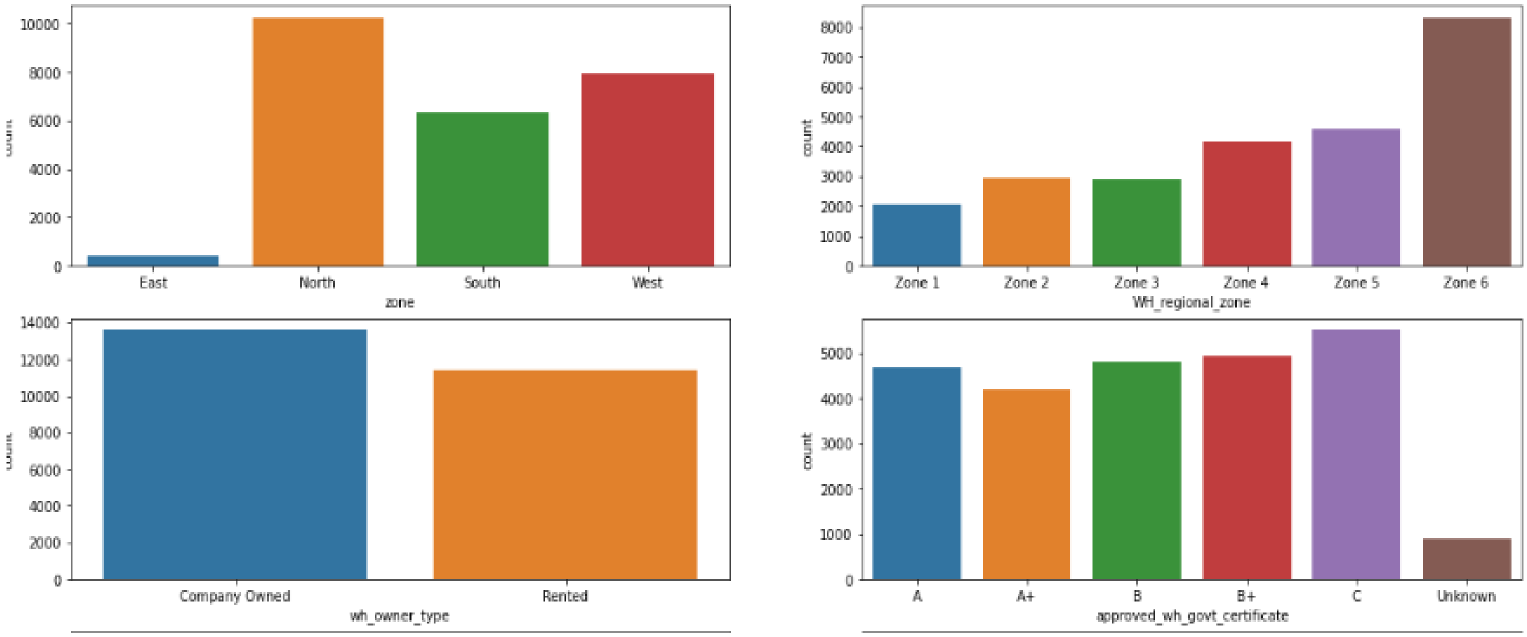


Figure 2 –Displot of Zones owner type

Bivariate Analysis

In Bivariate analysis we can compare 2 variables in a single plot to see the relation between both of them and see the necessary outcomes we can interpret through those variables, we can also include a third categorical variable in this with the help of hue to understand how different variables are separated with the help of hue kind. In this data set we have explored various scatterplot, count plot, barplot, boxplot to understand the data.

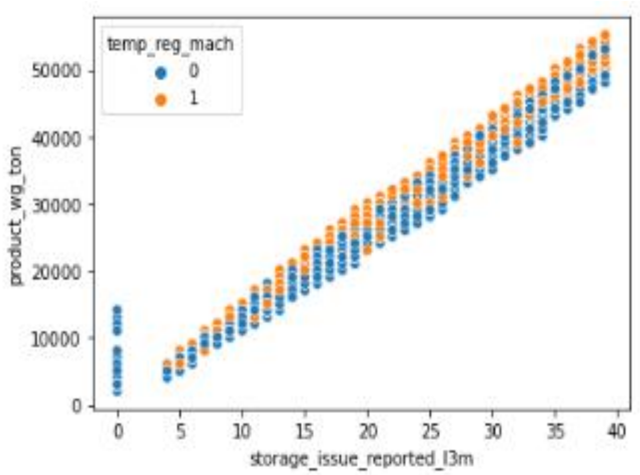


Figure 8 –Scatterplot for storage issue and product wg tonl

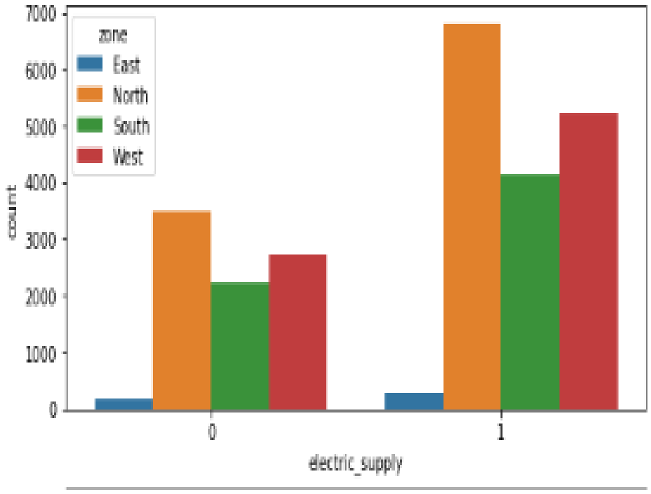


Figure 9 – countplot for zone and electricity supply

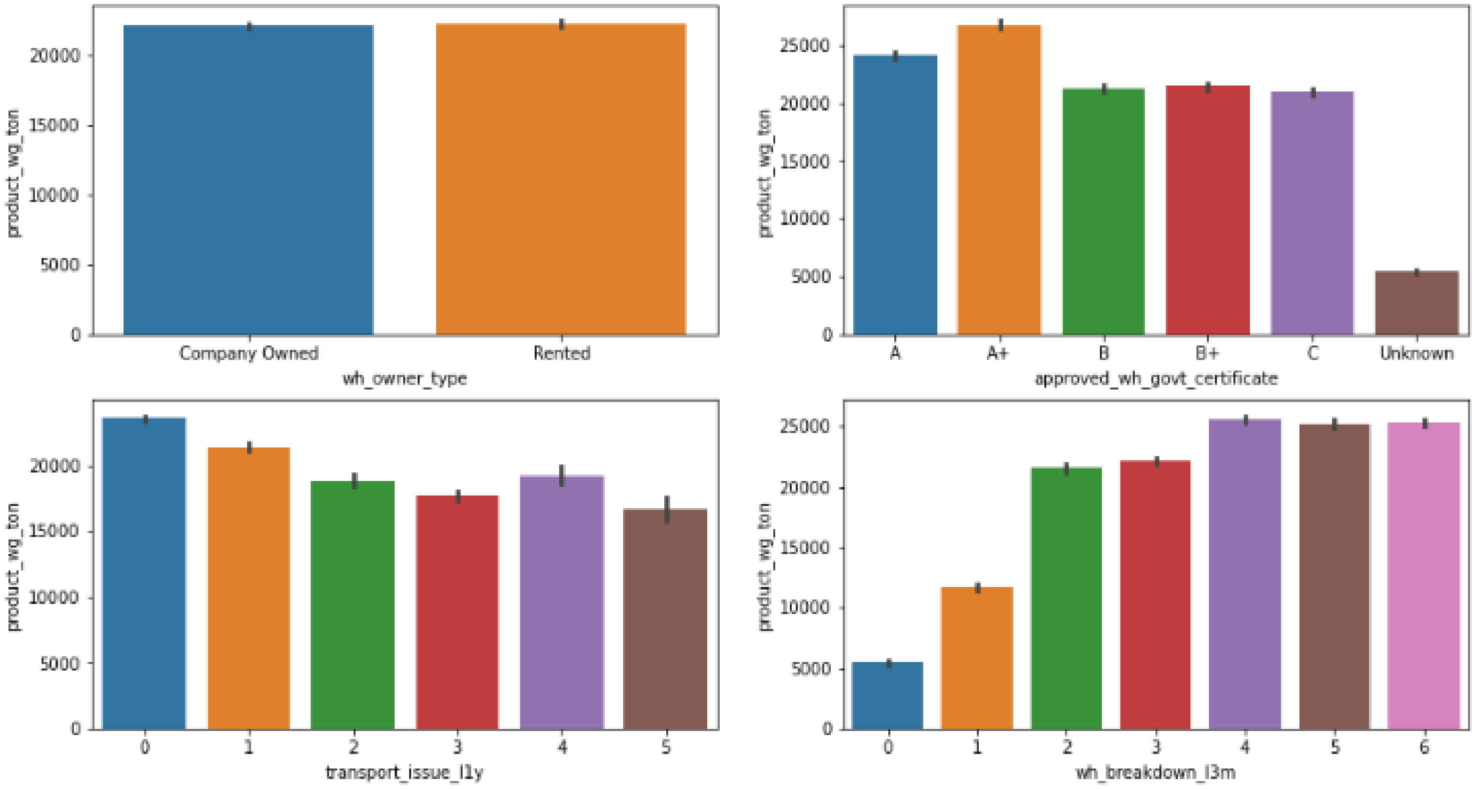


Figure 10 – countplot for wh owner type and transport issues etc.

Multivariate Analysis

In this analysis we can see all the numeric columns compared with each other at a time this can be done with two methods a Pair plot and a heat map. In this report we have included the heat map as it shows the correlation between each numeric variable with every other numeric variable available in the data set. Since the heatmap was very big it has been pasted in half and and to see the meaningfull output out of it.

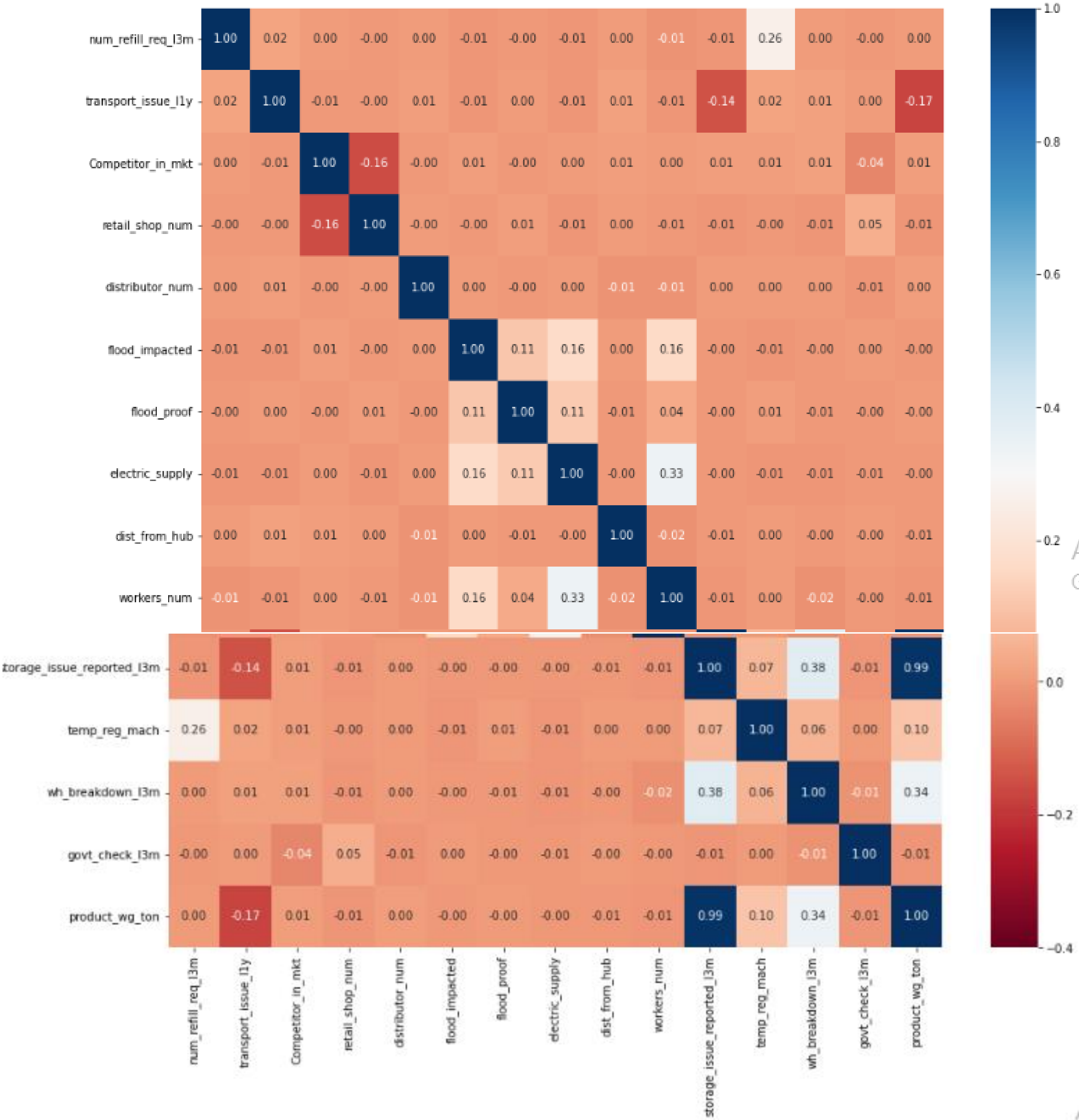


Figure 5 – Heatmap for all the variables.

Outlier Treatment

As discussed above the outlier treatments are changed to lower and upper Wisker value with the help of Inter quartile range as outliers are also part of the data set and can be helpful in creating meaning full outcomes from the data

Business insights from EDA

EDA – Exploratory Data Analysis this is a very beautiful and most widely used technique across all over the industry to understand each and every insides about the data.
Before any model buildings or working on any data set an EDA is very much required in order to get the overall glimpse of the data.

Key Insights from this Data set.

- We have looked for Duplicate records if anything is present in the data set which in this data set we were unable to find out any duplicate rows which is very much crucial part of EDA.
- Coming to the EDA graphs part which means univariate, bivariate and Multivariate analysis we have seen a lot of similarities and dissimilarities between the columns.
    - **In univariate analysis** we saw that most of the columns are normally distributed.
    - Retail Shops are mostly around 5000 shops for every warehouse and for some warehouses it even crosses 11000 which needs to be distributed equally so that the warehouse are not overburden with their inventory and work
    - In count plot we can clearly see that there is an **Imbalanced data** since there are very few warehouses in the urban areas and majority of the warehouses in the rural areas. The company should focus more on the urban areas because urban people have more spending power
    - We can also see an **Imbalance** in the zone wise distribution of the product as east zone has only 420 warehouses and north zone has more than 10000 warehouses.
    - **In Bivariate Analysis** we can see although urban area have such low amount of warehouses but the number of retailers between the urban and rural areas are almost equivalent and we also see their sale are also near to equivalent in fact urban is having a bit of more sales this says that the company's should spend more on urban part of the country as it has a higher potential market.
    - In the Flood impacted zones and flood proof zones also there is a big number difference the company should take appropriate steps to make the warehouses flood proofs at least in the flood impacted zones Also we see there is a significant difference in the sale of warehouses having temperature regulating machines which are helping the goods to sustain for longer times.
    - Although the count of C grade warehouse in govt_certification column is the most but its sale for the product is at least and the sale of A+ grade warehouses are the most therefore company should try and focus on these grades and take steps to improve overall quality of the warehouse.
    - In transport issue we can see that the warehouses having less number of transport issue are selling more products and warehouse having more transport issues are having less sale as well. This is a area of concern for the company and should focus immediately on this.
    - **In multi variate Analysis** we can see that storage issue is directly linked with our target variable as they are having a 0.99 of correlation between them
    - Most of the columns are having almost 0 or very close to zero that means almost no relation with each other which means there is not much of a multicollinearity in the data.
    - Other relations weather negative or positive lies between +0.2 to – 0.17 between the variables.

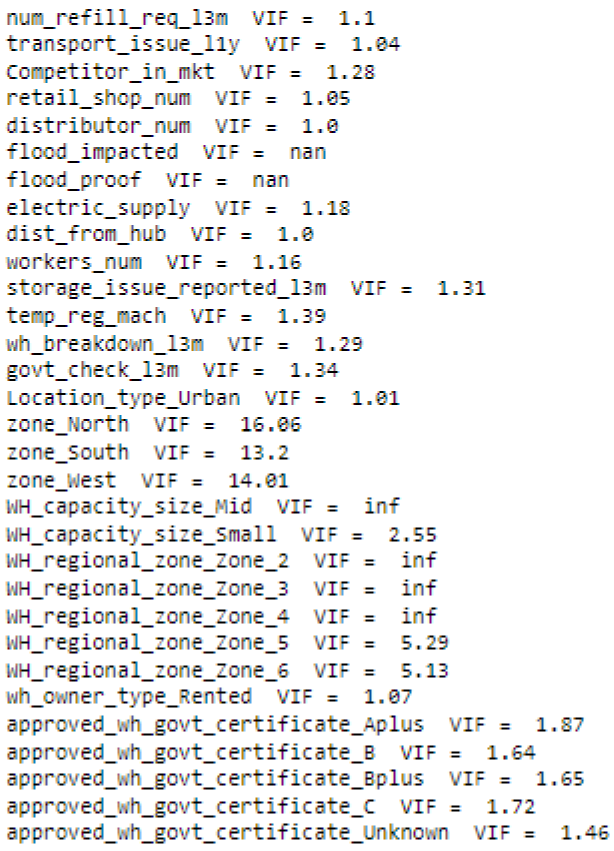
3. Model building and interpretation.

1. Linear Regression

Linear regression is a popular statistical modeling technique used to establish a relationship between a dependent variable and one or more independent variables. It assumes a linear relationship between the independent variables and the dependent variable. The goal of linear regression is to find the best-fitting line (or hyperplane) that minimizes the sum of the squared differences between the predicted and actual values.

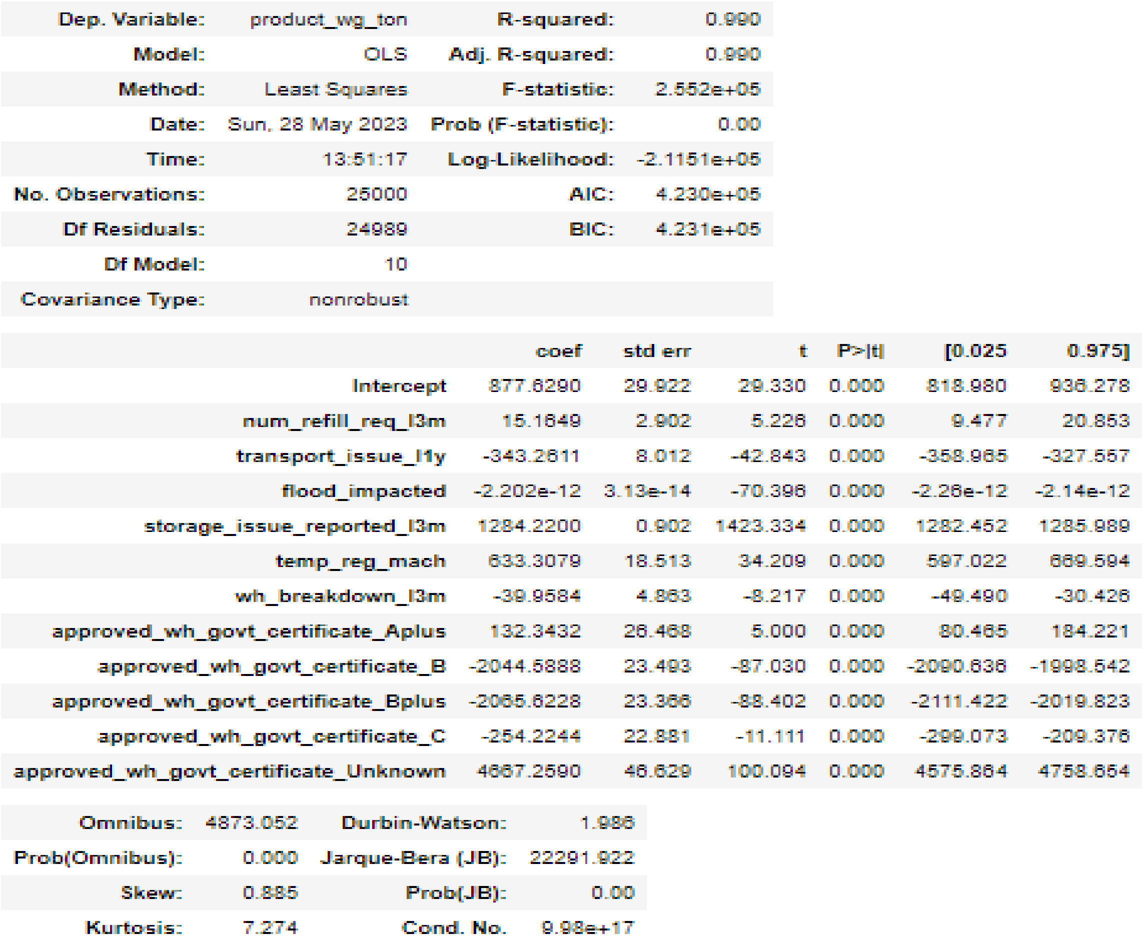
We have also splitted the data into Train and test models for 70:30 ratio to train and test the data based on the models we are going to perform
Then we took out the VIF values for our columns as in Linear regression problems we have to keep two criteria's in mind firstly we have to remove columns with VIF values more than 5 to avoid multicollinearity between the data and then we have to remove variables which are having alpha value more than 5%. By using minimum variables and making the best model is considered as the most useful and effective model.

Below are the initial VIF Values of the data in which we are going to remove the columns in later model building process.



```
num_refill_req_l3m  VIF =  1.1
transport_issue_l1y  VIF =  1.04
Competitor_in_mkt  VIF =  1.28
retail_shop_num  VIF =  1.05
distributor_num  VIF =  1.0
flood_impacted  VIF =  nan
flood_proof  VIF =  nan
electric_supply  VIF =  1.18
dist_from_hub  VIF =  1.0
workers_num  VIF =  1.16
storage_issue_reported_l3m  VIF =  1.31
temp_reg_mach  VIF =  1.39
wh_breakdown_l3m  VIF =  1.29
govt_check_l3m  VIF =  1.34
Location_type_Urban  VIF =  1.01
zone_North  VIF =  16.06
zone_South  VIF =  13.2
zone_West  VIF =  14.01
WH_capacity_size_Mid  VIF =  inf
WH_capacity_size_Small  VIF =  2.55
WH_regional_zone_Zone_2  VIF =  inf
WH_regional_zone_Zone_3  VIF =  inf
WH_regional_zone_Zone_4  VIF =  inf
WH_regional_zone_Zone_5  VIF =  5.29
WH_regional_zone_Zone_6  VIF =  5.13
wh_owner_type_Rented  VIF =  1.07
approved_wh_govt_certificate_Aplus  VIF =  1.87
approved_wh_govt_certificate_B  VIF =  1.64
approved_wh_govt_certificate_Bplus  VIF =  1.65
approved_wh_govt_certificate_C  VIF =  1.72
approved_wh_govt_certificate_Unknown  VIF =  1.46
```

In this intial model we also see a lot of variables having pvalue more than 5% therefore we remove all of them and present the final model in front of you. Then we have to apply linear Regression in this model to collect the Adjusted R^2 and RMSE from the data these are the indicators of model performance in case of Regression based models when the target variable is a continuous one.



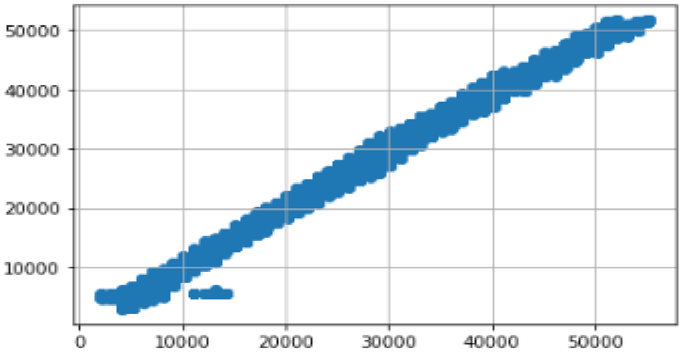
Dep. Variable:	product_wg_ton	R-squared:	0.990
Model:	OLS	Adj. R-squared:	0.990
Method:	Least Squares	F-statistic:	2.552e+05
Date:	Sun, 28 May 2023	Prob (F-statistic):	0.00
Time:	13:51:17	Log-Likelihood:	-2.1151e+05
No. Observations:	25000	AIC:	4.230e+05
Df Residuals:	24989	BIC:	4.231e+05
Df Model:	10		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	877.6290	29.922	29.330	0.000	818.980	936.278
num_refill_req_l3m	15.1649	2.902	5.226	0.000	9.477	20.853
transport_issue_l1y	-343.2611	8.012	-42.843	0.000	-358.965	-327.557
flood_impacted	-2.202e-12	3.13e-14	-70.396	0.000	-2.26e-12	-2.14e-12
storage_issue_reported_l3m	1284.2200	0.902	1423.334	0.000	1282.452	1285.989
temp_reg_mach	633.3079	18.513	34.209	0.000	597.022	669.594
wh_breakdown_l3m	-39.9584	4.863	-8.217	0.000	-49.490	-30.426
approved_wh_govt_certificate_Aplus	132.3432	26.468	5.000	0.000	80.465	184.221
approved_wh_govt_certificate_B	-2044.5888	23.493	-87.030	0.000	-2090.636	-1998.542
approved_wh_govt_certificate_Bplus	-2065.6228	23.366	-88.402	0.000	-2111.422	-2019.823
approved_wh_govt_certificate_C	-254.2244	22.881	-11.111	0.000	-299.073	-209.376
approved_wh_govt_certificate_Unknown	4667.2590	46.629	100.094	0.000	4575.864	4758.654

Omnibus:	4873.052	Durbin-Watson:	1.986
Prob(Omnibus):	0.000	Jarque-Bera (JB):	22291.922
Skew:	0.885	Prob(JB):	0.00
Kurtosis:	7.274	Cond. No.	9.98e+17

This is the final model build on linear regression. It is the model number 21 which means a total of 20 variables were removed from the data which were multicollinear and having VIF value more than 5 or alpha value more than 0.05. Multicollinearity means major part of that variable is been explained by another variable which is still present in the model and if we remove that variable as well it is not going to show much of a difference in the accuracy score which is Adjusted R^2 in this case. Surprisingly we have removed 20 variables in this model but the adjusted R^2 has not moved much and it shows a same result atleast upto 3 decimals in the model.

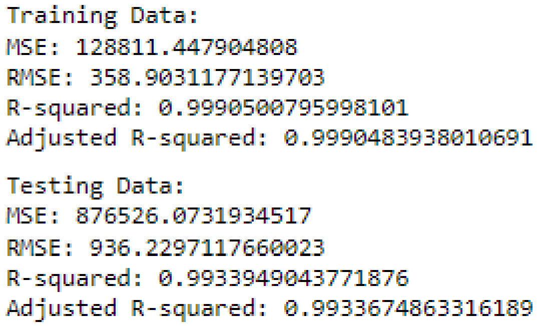
In this case we see a accuracy score that is adjusted R^2 of 99.03% and the RMSE of both train and test data are 1153.848 for train data and 1118.162 for test data.



This Scatter plot almost the best fit line of the remaining variables and data points and if we see the data are well aligned with the best fit line.

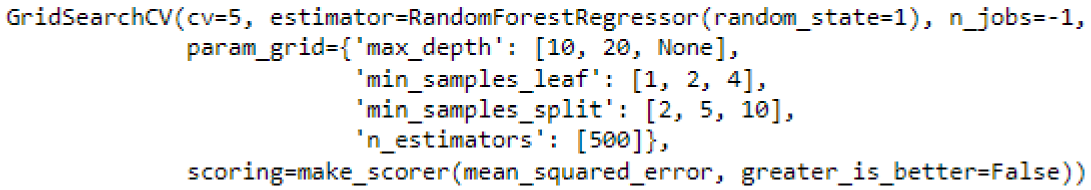
2. Random Forest Regressor.

The Random Forest Regressor is a popular machine learning algorithm that belongs to the ensemble learning family. It is based on the concept of decision trees and combines multiple decision trees to make predictions.
A Random forest model is a data mining technique in which it build on a model of multiple trees to build a forest of data and then average of all the trees are counted to get the final output.



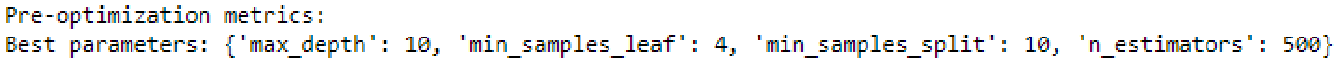
```
Training Data:
MSE: 128811.447904808
RMSE: 358.9031177139703
R-squared: 0.9990500795998101
Adjusted R-squared: 0.9990483938010691

Testing Data:
MSE: 876526.0731934517
RMSE: 936.2297117660023
R-squared: 0.9933949043771876
Adjusted R-squared: 0.9933674863316189
```

This is the output for training and testing data in random forest with 500 estimaters taken into consideration.
In the above estimater as we see the RMSE values are way too different for both training and testing data therefore there is also a concept of grid search in this. In grid search the Algorithm inside the machine decides automatically the number of estimaters which should be considered then number rows in which the data will have to split every thing is calculated by the algorithm and then it gives output as best estimators suitable for the data



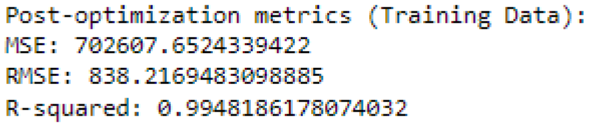
```
GridSearchCV(cv=5, estimator=RandomForestRegressor(random_state=1), n_jobs=-1,
             param_grid={'max_depth': [10, 20, None],
                         'min_samples_leaf': [1, 2, 4],
                         'min_samples_split': [2, 5, 10],
                         'n_estimators': [500]},
             scoring=make_scorer(mean_squared_error, greater_is_better=False))
```

This many Estimators and data splitting is been taken into consideration by the model.

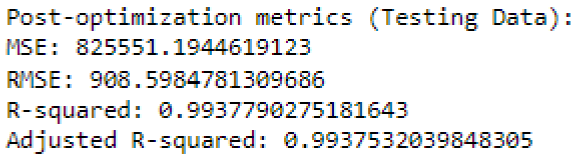


```
Pre-optimization metrics:
Best parameters: {'max_depth': 10, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 500}
```

The model has chosen this as the best split for the data.
Then we must fit our train and test model into the grid search with its split to get the best possible outcome from the data.



```
Post-optimization metrics (Training Data):
MSE: 702607.6524339422
RMSE: 838.2169483098885
R-squared: 0.9948186178074032
```

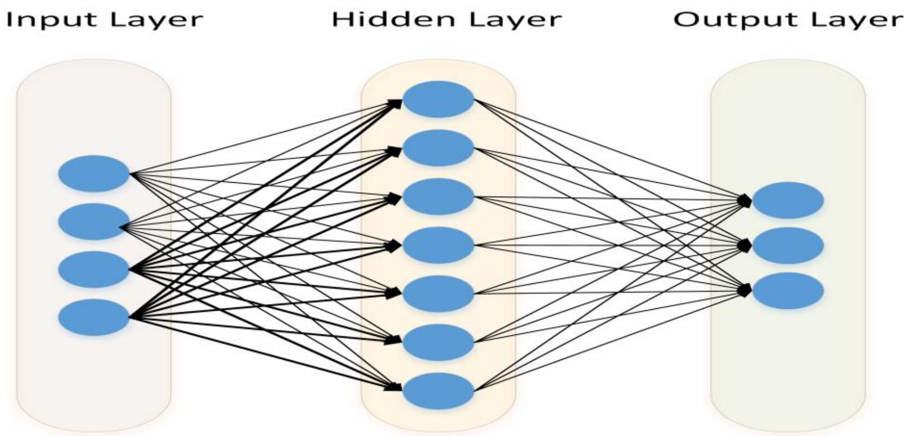


```
Post-optimization metrics (Testing Data):
MSE: 825551.1944619123
RMSE: 908.5984781309686
R-squared: 0.9937790275181643
Adjusted R-squared: 0.9937532039848305
```

In this model after searching from grid search as we see the RMSE values came closer to each other which means the model is more reliable now also the adjusted R2 has increased and came closer to R2 value of training data which is a good sign because when the training and testing are more nearer to each other that means it is a good and reliable model.

3. **Neural Networks.**

Neural networks, also known as artificial neural networks or simply neural nets, are a class of machine learning models inspired by the structure and functioning of the human brain. They are widely used for various tasks, including classification, regression, pattern recognition, and decision-making.



Here are some key aspects of neural networks:

Architecture: A neural network is composed of interconnected nodes called neurons organized in layers. The three main types of layers are the input layer, hidden layer(s), and output layer. The input layer receives the input data, the hidden layer(s) process and transform the information, and the output layer provides the final predictions or results.

Neurons and Activation Functions: Each neuron in a neural network receives inputs, performs a computation, and produces an output. Neurons apply an activation function to the weighted sum of their inputs, introducing non-linearity to the model. Common activation functions include sigmoid, tanh, and ReLU (Rectified Linear Unit).

Weighted Connections: Neurons in one layer are connected to neurons in the subsequent layer through weighted connections. These weights determine the strength and importance of the connections. During training, the network adjusts these weights to optimize the model's performance.

Forward Propagation: In a neural network, forward propagation refers to the process of passing input data through the network, layer by layer, to obtain predictions or outputs. Each layer performs a computation based on its weights, activation function, and inputs from the previous layer.

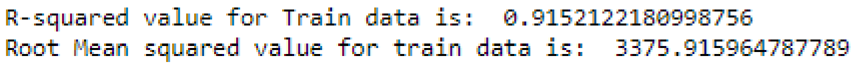
Backpropagation: Backpropagation is a key algorithm for training neural networks. It calculates the gradients of the model's parameters (weights and biases) with respect to a loss function, allowing the model to adjust its weights and improve its performance. It iteratively updates the weights by propagating the error from the output layer back to the input layer.

Deep Learning: Deep learning is a subset of neural networks that consists of models with multiple hidden layers. Deep neural networks (DNNs) are capable of learning hierarchical representations of data and have achieved remarkable success in various domains, including image recognition, natural language processing, and speech recognition.

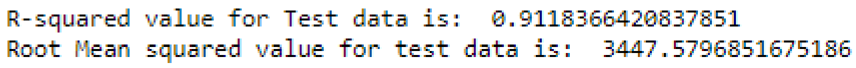
Hyperparameters: Neural networks have various hyperparameters that need to be set before training, such as the number of layers, number of neurons per layer, learning rate, regularization parameters, and batch size. Proper tuning of these hyperparameters is crucial for optimal performance.

Neural networks are known for their ability to learn complex patterns and relationships in data, making them highly flexible and powerful models. However, they also require substantial computational resources and large amounts of data for training. Advances in hardware and algorithms have fueled the rapid growth and success of neural networks, particularly in the field of deep learning.

Initial Result of training and testing data through neural networks.

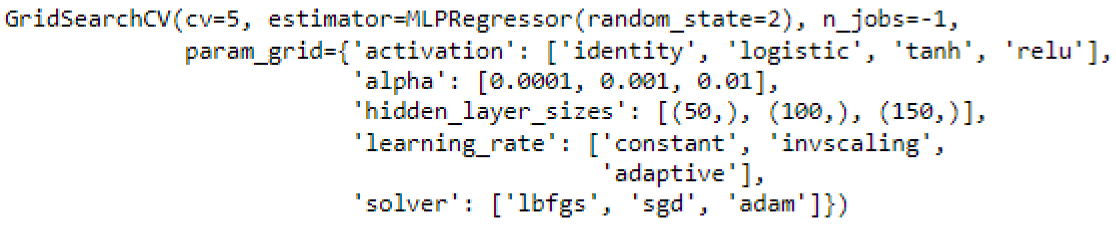


```
R-squared value for Train data is:  0.9152122180998756
Root Mean squared value for train data is:  3375.915964787789
```



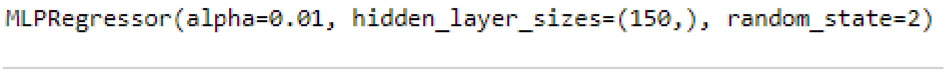
```
R-squared value for Test data is:  0.9118366420837851
Root Mean squared value for test data is:  3447.5796851675186
```

In this model also we search through grid search about their actual potential for splitting of the data into different layers, its learning rate, which method it should solve etc.



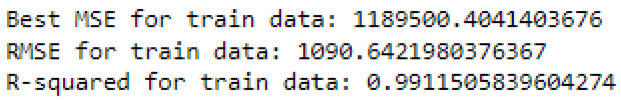
```
GridSearchCV(cv=5, estimator=MLPRegressor(random_state=2), n_jobs=-1,
             param_grid={'activation': ['identity', 'logistic', 'tanh', 'relu'],
                         'alpha': [0.0001, 0.001, 0.01],
                         'hidden_layer_sizes': [(50,), (100,), (150,)],
                         'learning_rate': ['constant', 'invscaling',
                                           'adaptive'],
                         'solver': ['lbfgs', 'sgd', 'adam']})
```

This is the parameters in which grid search searches the best possible result for the data.

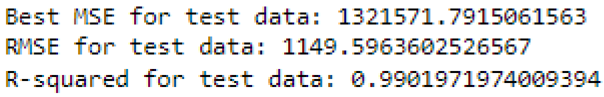


```
MLPRegressor(alpha=0.01, hidden_layer_sizes=(150,), random_state=2)
```

The best fit parameter which the model has suggested for this particular data. Now we have to fit this into training and testing of the data and get possible outcome out of it.



```
Best MSE for train data: 1189500.4041403676
RMSE for train data: 1090.6421980376367
R-squared for train data: 0.9911505839604274
```

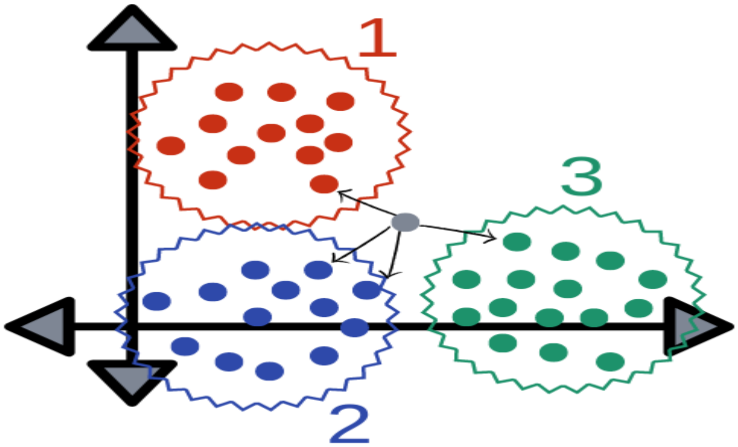


```
Best MSE for test data: 1321571.7915061563
RMSE for test data: 1149.5963602526567
R-squared for test data: 0.9901971974009394
```

As we see the R2 value has increased exponentially when we have taken the estimaters from grid search. It has increased by 8% approximately which was initially 91% approximately and then after the use of grid search best estimators the R2 has increased to 99% which means the model is more reliable now.

4. **K-Nearest Neighbors Regressor.**

The K-nearest neighbors (KNN) regressor is a type of supervised learning algorithm used for regression tasks. It predicts the value of a target variable based on the values of its K nearest neighbors in the feature space.



This is an image showing how the KNN works by making a group within its nearest neighbours

Here's an overview of the K-nearest neighbors regressor:

Nearest Neighbor Concept: The KNN algorithm is based on the idea that similar instances (or neighbors) in the feature space tend to have similar target values. It assumes that if a new data point is close to its K nearest neighbors, its target value will likely be similar to those neighbors.

Training Phase: During the training phase of the KNN regressor, the algorithm simply stores the feature values and corresponding target values of the training data. No explicit model is built or learned.

Prediction Phase: When making predictions with KNN regressor, the algorithm calculates the distance between the new data point and all the training instances. The K nearest neighbors are selected based on the shortest distances.

Choosing K: The value of K represents the number of nearest neighbors to consider when making predictions. It is an important hyperparameter that needs to be tuned. A small value of K can make the model sensitive to noise, while a large value of K may lead to oversmoothing and reduced sensitivity to local patterns.

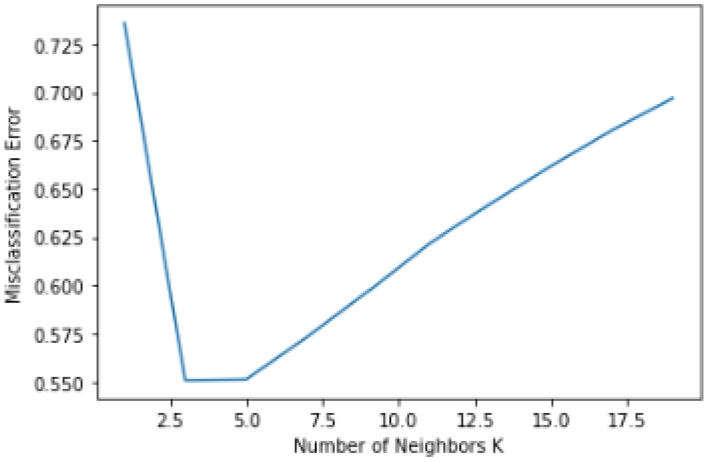
Distance Metric: The choice of distance metric, such as Euclidean distance or Manhattan distance, determines how the algorithm measures the proximity between instances in the feature space. The appropriate distance metric depends on the nature of the data and the problem at hand.

Prediction Calculation: For regression tasks, the predicted value for the new data point is often calculated as the mean or median of the target values of its K nearest neighbors. Weighted averaging can also be used, where closer neighbors have more influence on the prediction.

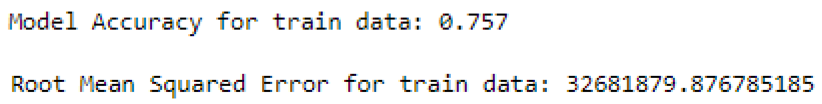
Scaling and Preprocessing: Feature scaling and preprocessing techniques, such as normalization or standardization, can be important when working with KNN regressor. Rescaling the features to a similar scale can prevent variables with larger ranges from dominating the distance calculations.

KNN regressor is a simple yet effective algorithm for regression tasks. However, it can be computationally expensive when dealing with large datasets, as it requires calculating distances to all training instances. Additionally, choosing an appropriate value for K and selecting the right distance metric are crucial for obtaining accurate predictions.

- In this First we have to find out the number of nearest neighbors that we have to form to perform the model.
  The best way to find this out is through misclassification errors where the minimum number of errors are present that is best suitable for model building.

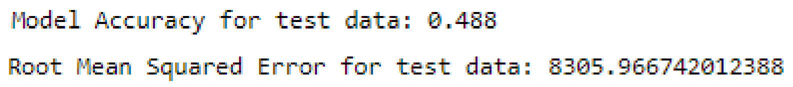


This is the graph showing multiple Misclassification error in which we can see that from k value 3 to 5 has the least number of errors which are present so we took 3 in our model as the number of K to perform the model.



```
Model Accuracy for train data: 0.757

Root Mean Squared Error for train data: 32681879.876785185
```

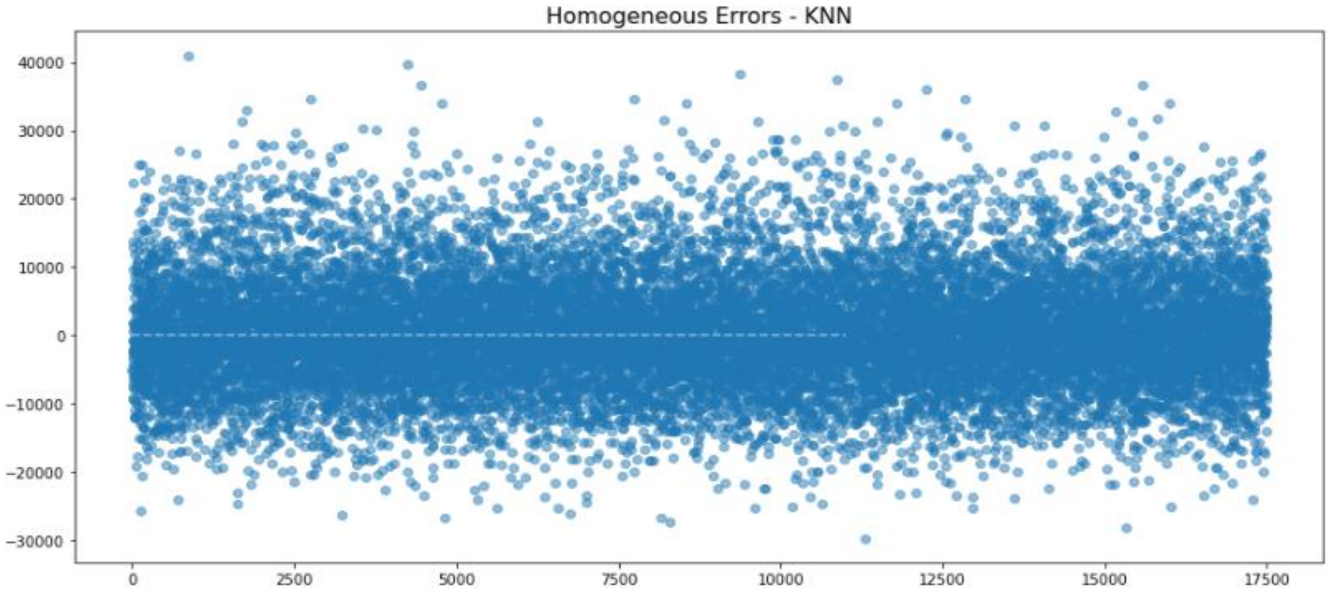
This model is the worst so far in comparison to we have saw the other models in this project in this the model accuracy score that means the Adjusted r2 is only 75% and also the root mean square is way too high which is not considered as a good model



```
Model Accuracy for test data: 0.488
Root Mean Squared Error for test data: 8305.966742012388
```

In this the test data shows only a 48% of accuracy and also there is a huge difference between the

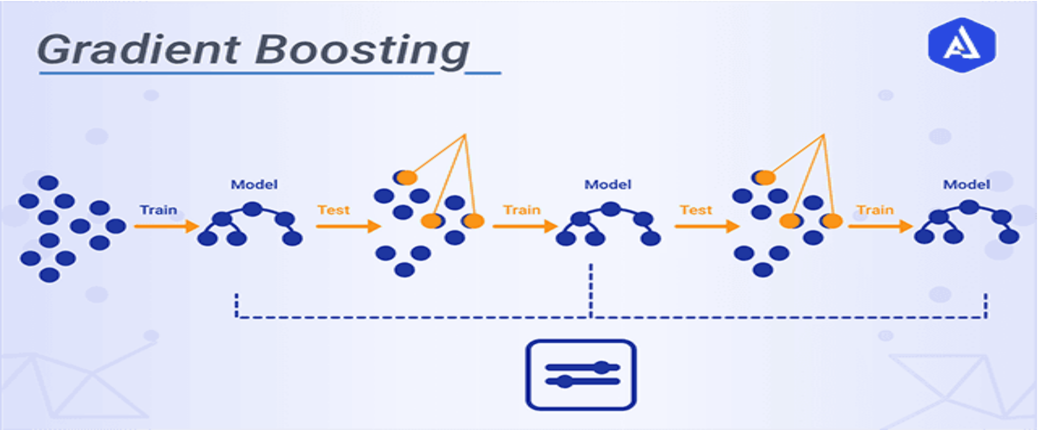
root mean square between the training and testing of the data. Also the difference between the training and testing of the model shows more than 25% of gap between them which means the model is over fitting in this case.
Therefore this model is not at all recommended in this data.



This is the Homogeneous Error for the data as we see the data points are verymuch cluttered and far away from the best fit line therefore it is not showing such accurate results.

5. **Gradient Boosting.**

Gradient Boosting is a machine learning technique that belongs to the ensemble learning family. It combines multiple weak learners, typically decision trees, to create a strong predictive model. It is known for its effectiveness in handling complex datasets and achieving high predictive accuracy.



Here's an overview of Gradient Boosting:

Boosting Concept: The key idea behind Gradient Boosting is to build a strong model by iteratively adding weak learners in a sequential manner. Each weak learner is trained to correct the mistakes made by the previous learners, with a focus on the instances that were not well predicted.

Gradient Descent Optimization: Gradient Boosting minimizes a loss function by iteratively updating the model's parameters. It uses gradient descent optimization to find the direction and magnitude of updates that reduce the loss. The loss gradient is computed based on the difference between the predicted and actual values.

Decision Tree Base Learners: In Gradient Boosting, decision trees are commonly used as base learners. Decision trees are trained on the residuals or errors made by the previous learners. Each subsequent tree is designed to reduce the remaining errors, leading to a gradual improvement in predictions.

Ensemble of Weak Learners: The final prediction in Gradient Boosting is a weighted sum of the predictions from all the weak learners. Each learner contributes to the final prediction based on its individual weight, which is determined by its performance and the learning rate hyperparameter.

Regularization: Gradient Boosting models are prone to overfitting. To address this, regularization techniques are often employed, such as limiting the depth of the trees, applying shrinkage to the learning rate, and introducing randomness through subsampling of the data or features.

Hyperparameters: Gradient Boosting models have various hyperparameters to be tuned, including the number of boosting iterations, learning rate, maximum tree depth, subsampling rate, and regularization parameters. Proper tuning of these hyperparameters is important for achieving optimal performance.

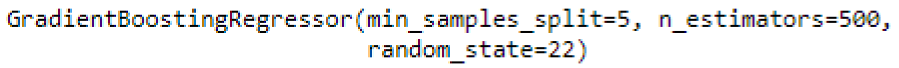
Feature Importance: Gradient Boosting models can provide insights into feature importance, indicating the relative importance of features in making predictions. This information can be valuable for feature selection and understanding the underlying relationships in the data.

Gradient Boosting is widely used in various domains and has achieved great success in machine learning competitions and real-world applications. Popular implementations of Gradient Boosting include XGBoost, LightGBM, and CatBoost, which provide optimized algorithms and additional features for improved performance and efficiency.

In this technique also we find the best parameters to fit into the training and testing of the data in order to get the best output out of it.

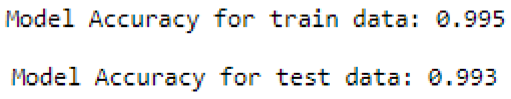
```
gbr_param = {
      "n_estimators"     : 500,
      "max_depth"        : 3,
      "min_samples_split" : 5,
      "learning_rate"    : 0.1,
      "criterion"        : 'friedman_mse',
      "loss"             : 'squared_error',
          }
```

These are the parameters at which the gradient boosting method works on and finds out the best parameter for our data.



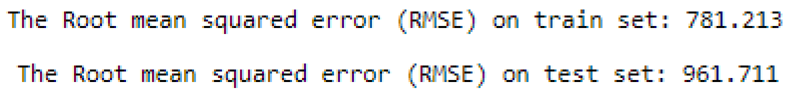
```
GradientBoostingRegressor(min_samples_split=5, n_estimators=500,
                          random_state=22)
```

This is the best parameter which the algorithm has found out.



```
Model Accuracy for train data: 0.995

 Model Accuracy for test data: 0.993
```

It shows a very high and very close accuracy scores between training and testing data which means this is a reliable model to see.



```
The Root mean squared error (RMSE) on train set: 781.213

 The Root mean squared error (RMSE) on test set: 961.711
```

These are the root mean square error given by the model which is also very good and comparable this is by far one of the best model we have seen.

4. Final recommendation

	Adjusted R^2		Root mean squared error (RMSE)		
Models	Training	Testing	Training	Testing	Rank of models
Linear Regression	99.03	99	1153.848	1118.162	4
Random forest Regressor	99.48	99.37	838.217	908.598	2
Neural Networks	99.11	99.01	1090.642	1149.596	3
K-Nearest Neighbors Regressor	75.7	48.8	32681879	8305	5
Gradient Boosting	99.55	99.31	781.213	961.711	1

In this we found out That Gradient Boosting is the best model which has the lowest Mean square error in training and also highest accuracy score.
We can also take Random forest model into consideration since in test data it is maximum close to the training data and also the lowest but the reason why it is been given 2nd rank because it shows more mean square error in the training data.
The K-nearest neighbor is the worst model and cannot be considered for model building of the data. It shows an overfitting in the models and also highest RMSE in the data.

Based on these models we can recommend various insights to the business:

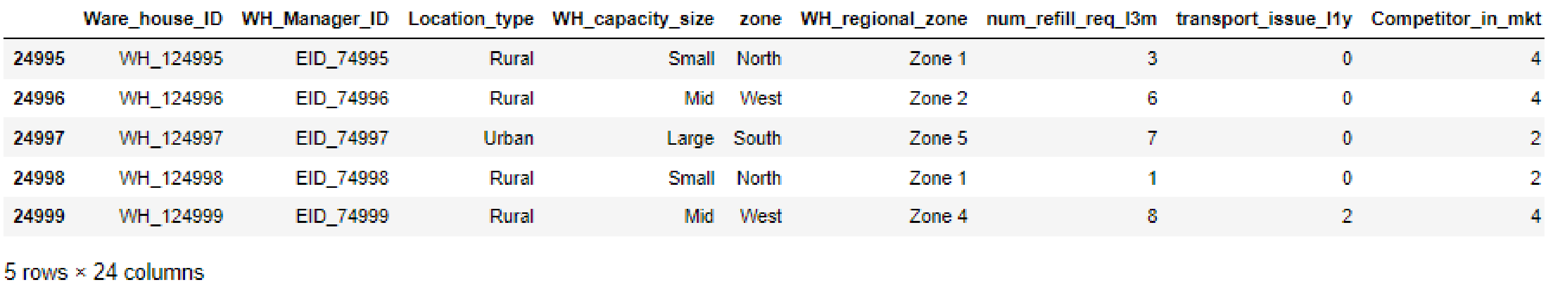
- There are some variables which are very much necessary for the business and the company needs to focus on those variables as we saw in the linear regression model.
- Government Certification plays a vital role in the sale of products of the business therefore we recommend to increase the rankings in the certificates as we see better the rank better is the sale for that particular warehouse.
- **Inventory Management** – The company needs to manage its inventory and need to share more inventories only on those warehouses those who actually need it and where the sale is

more and should not keep more dead stock at the warehouses which has less demand this will destroy the product the company can circulate that particular product to different warehouses where the demand is more this way wastage will be reduced for the product.

- **Demand Management** – The company needs to fulfill the demand according to the data present like we see there are few number of stores in the rural areas but there is a huge demand of noodles from that area where as people from rural area are not consuming noodels upto that extend so company needs to focus on rural areas much strongly.
- **Cost Management** – The company needs to manage its unnecessary cost which it is giving like for theft of goods, storage issues in the warehouses, number of workers working in a particular warehouse, flood impacted zones need to have special care and appropriate measures should be taken so that the product doesn’t get destroyed. The company should be able to manage the storage issues which are reported and we see that north zone has the highest storage issues so the company needs to take appropriate steps like visiting the stores every quarter and extracting their real problems. Transport issues is also a major problem for the Warehouses nearly 40% of the warehouse all over have reported issues in the transport and these areas are also sensitive areas so we should try any other root of transport for these areas if possible or we can also take help of law governing bodies this will indeed help the company to reduce it costs. One more point is the temperature control machines we have been seeing that in some areas these temperature control machines are very much effective and these has increased the durability of the goods, this can be put where the stocks take more time then usual to be cleared up in order to increase the life of the product
- **Warehouse Management** – In this we have to solve the breakdown issue of the warehouses that they are addressing we have seen that in some of the warehouses the breakdown issue has even gown up to 6 complains this will indeed cause trouble to the warehouses to run their business smoothly. Also many of the warehouses are having storage issues which is been raised in every three months we have to identify the most common problem with storage issue and make more storage hubs for that areas so that the warehouse is not been affected with this issue.
- **Transport Management** – We can see some of the warehouses are very far away from the retailers which costs extra money and extra time and also due to which more and more transport issue is arriving if we open more warehouses nearer to the retailers wherever possible then it can reduce the transport issues and overall cost for the business.
- **Operational Management** – The company has to hire more managers in order to smooth flow of these issues and a monthly visits in this warehouses because these are the face of the business and sends our products to the consumers if the retailers are not happy and not working with full potential for us then it will be very difficult to grow the business exponentially.

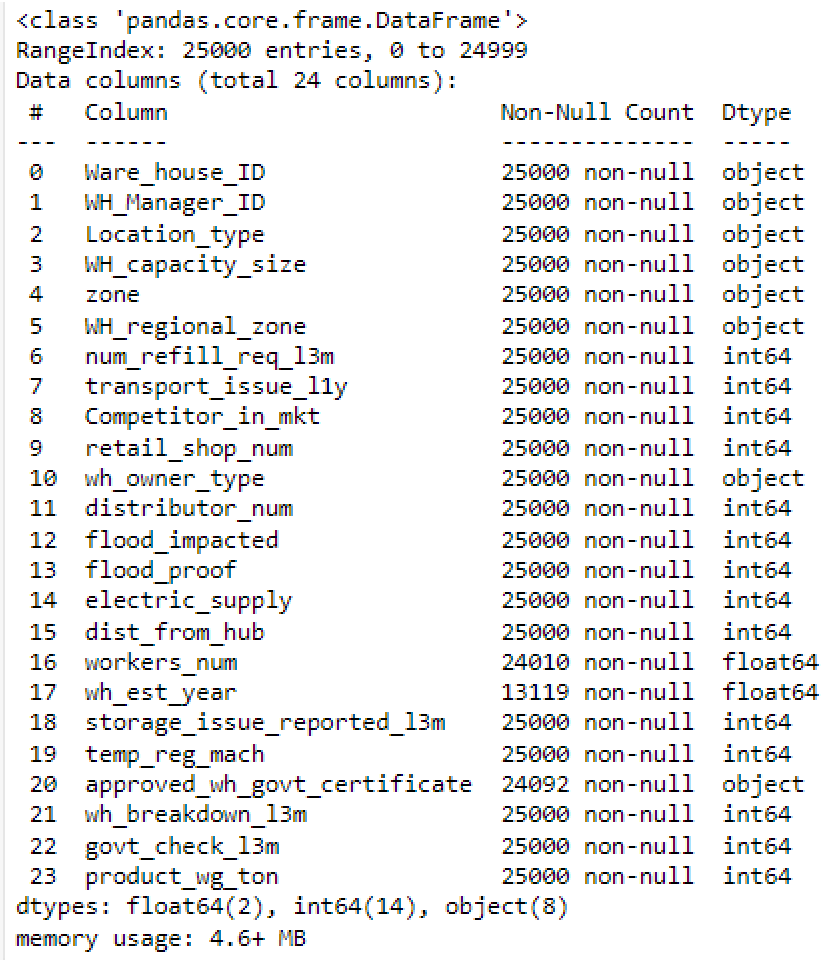
If company follow these points or recommendations and implement it in their business it can see a change in the overall growth and smooth functioning in the business day to day operations as well. These points can add a lot of value to the structure of the company and when the business reaches a certain scale a proper structured manner growth is very much crucial for its successful running otherwise these small ignorance can cost the business heavily in the future.

APPENDIX

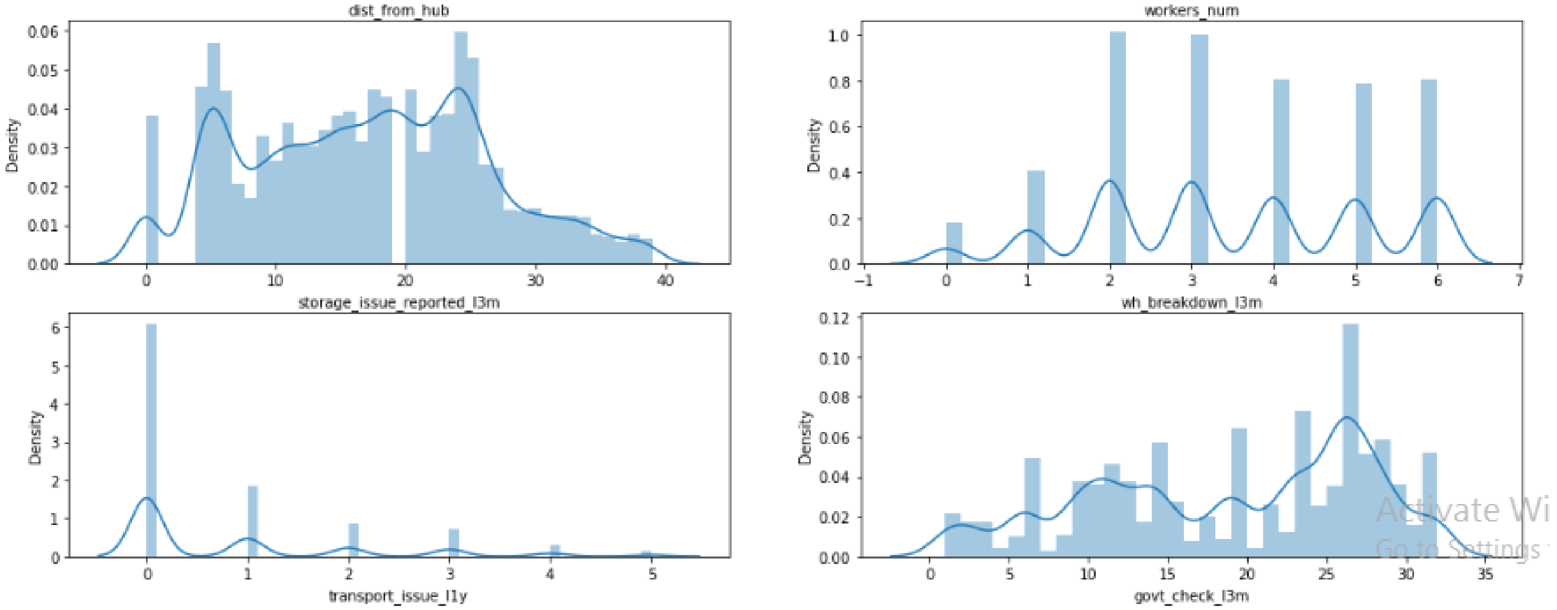


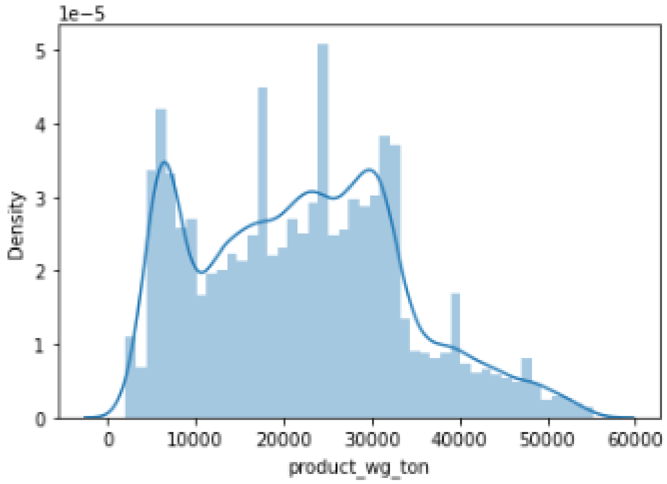
	Ware_house_ID	WH_Manager_ID	Location_type	WH_capacity_size	zone	WH_regional_zone	num_refill_req_l3m	transport_issue_l1y	Competitor_in_mkt
24995	WH_124995	EID_74995	Rural	Small	North	Zone 1	3	0	4
24996	WH_124996	EID_74996	Rural	Mid	West	Zone 2	6	0	4
24997	WH_124997	EID_74997	Urban	Large	South	Zone 5	7	0	2
24998	WH_124998	EID_74998	Rural	Small	North	Zone 1	1	0	2
24999	WH_124999	EID_74999	Rural	Mid	West	Zone 4	8	2	4

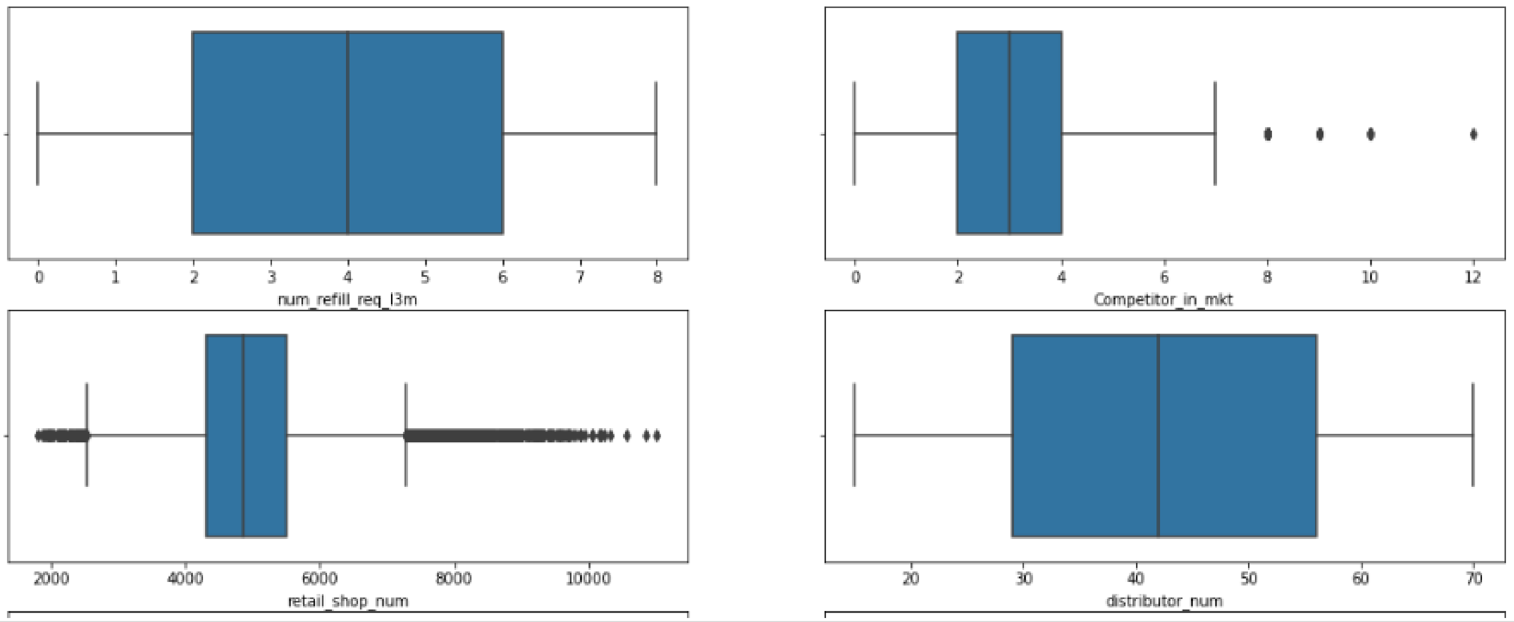
5 rows × 24 columns

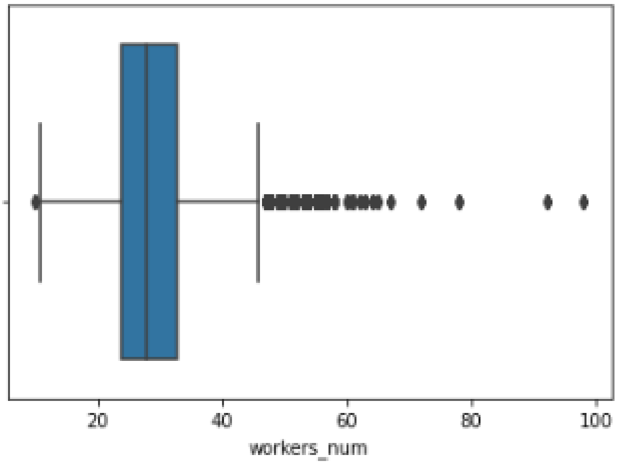


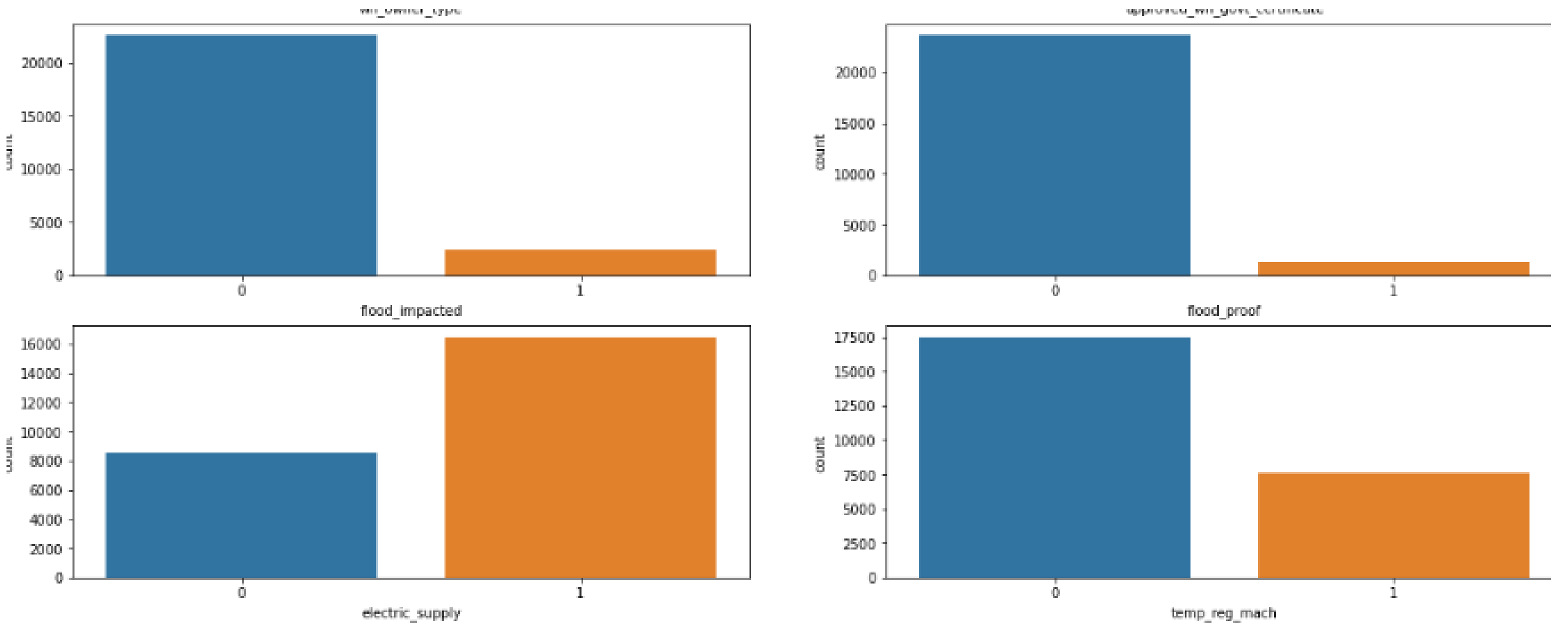
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25000 entries, 0 to 24999
Data columns (total 24 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   Ware_house_ID                 25000 non-null  object 
 1   WH_Manager_ID                 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
```

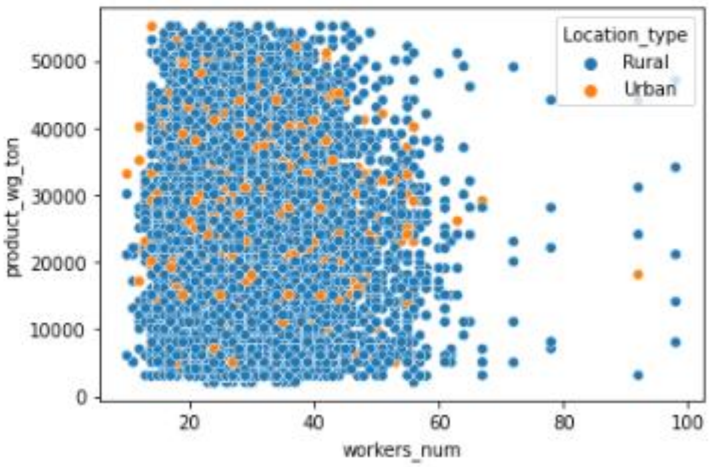

dist_from_hub
Density
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0
10
20
30
40
storage_issue_reported_l3m
workers_num
Density
1.0
0.8
0.6
0.4
0.2
0.0
−1
0
1
2
3
4
5
6
7
wh_breakdown_l3m
Density
6
5
4
3
2
1
0
0
1
2
3
4
5
transport_issue_l1y
Density
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
5
10
15
20
25
30
35
govt_check_l3m
Activate Wi
Go to Settings

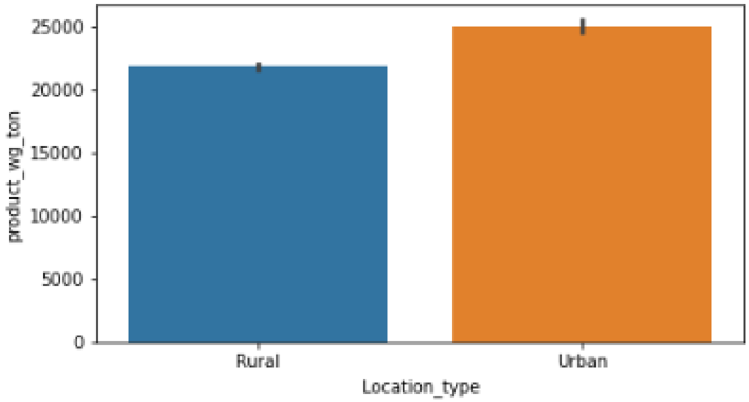

1e−5
Density
5
4
3
2
1
0
0
10000
20000
30000
40000
50000
60000
product_wg_ton

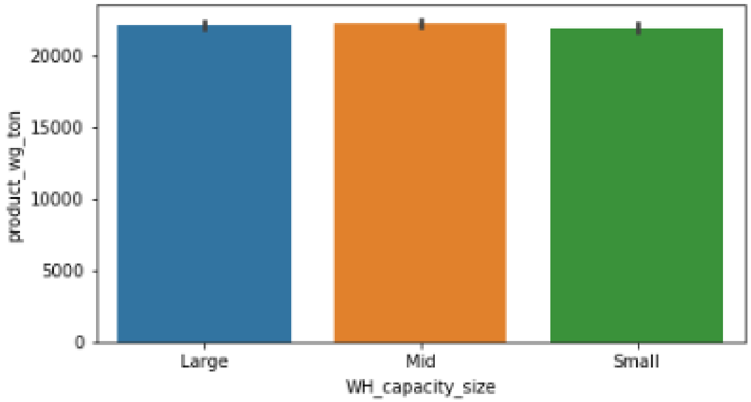

0
1
2
3
4
5
6
7
8
num_refill_req_l3m
0
2
4
6
8
10
12
Competitor_in_mkt
2000
4000
6000
8000
10000
retail_shop_num
20
30
40
50
60
70
distributor_num

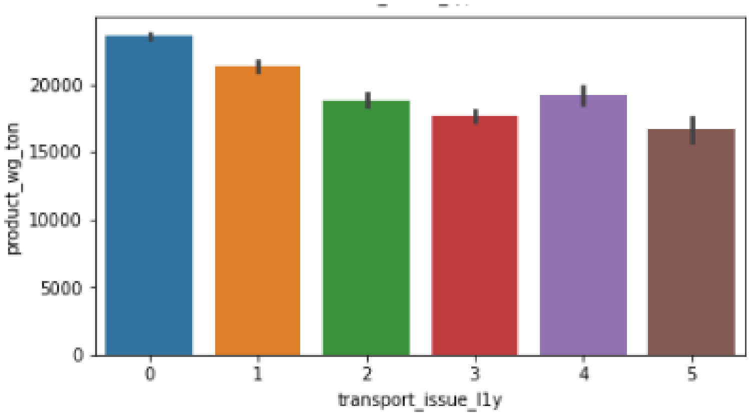

20
40
60
80
100
workers_num

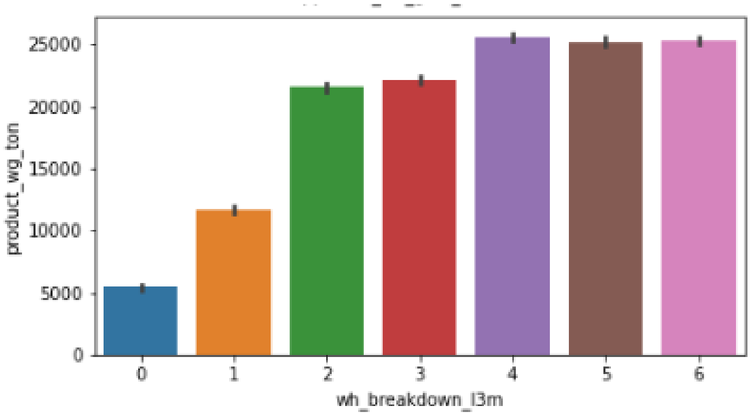

wh_owner_type
approved_wh_govt_certificate
count
20000
15000
10000
5000
0
0
1
flood_impacted
flood_proof
16000
14000
12000
10000
8000
6000
4000
2000
17500
15000
12500
7500
2500
electric_supply
temp_reg_mach

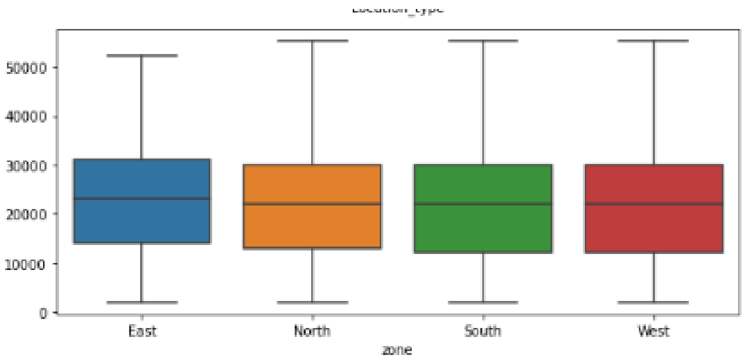

Location_type
Rural
Urban
product_wg_ton
50000
40000
30000
20000
10000
0
20
40
60
80
100
workers_num

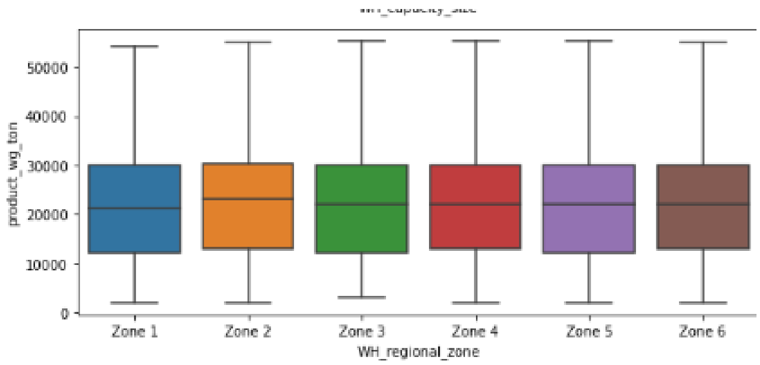

25000
20000
15000
10000
5000
0
product_wg_ton
Rural
Urban
Location_type

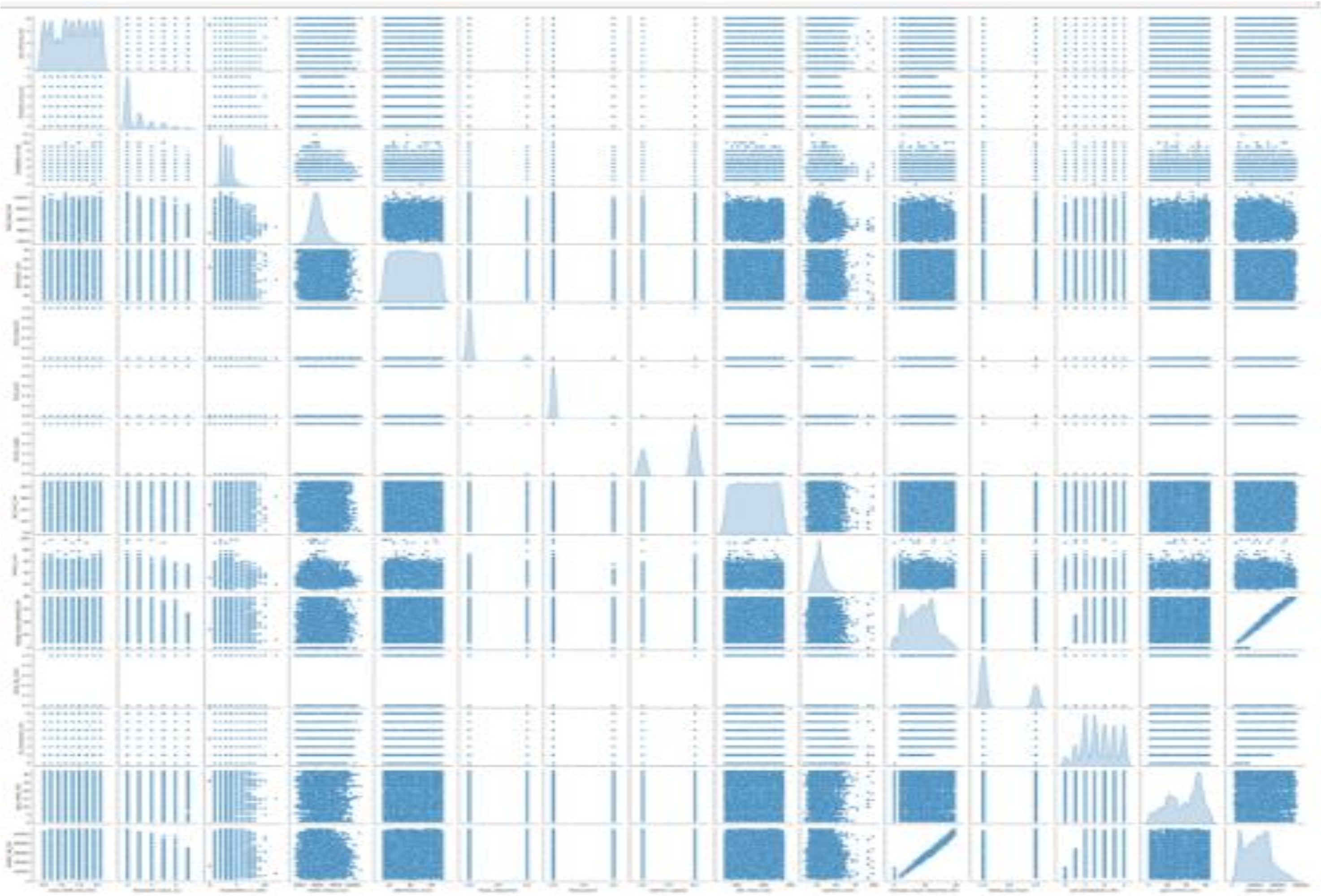

20000
15000
10000
5000
0
product_wg_ton
Large
Mid
Small
WH_capacity_size

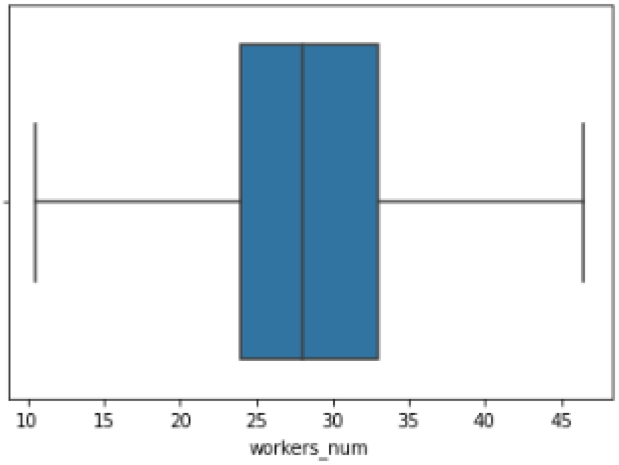

20000
15000
10000
5000
0
product_wg_ton
0
1
2
3
4
5
transport_issue_l1y

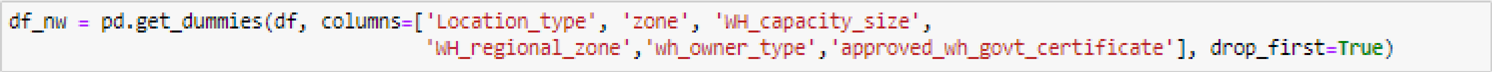

25000
20000
15000
10000
5000
0
product_wg_ton
0
1
2
3
4
5
6
wh_breakdown_l3m


Location_type
50000
40000
30000
20000
10000
0
East
North
South
West
zone

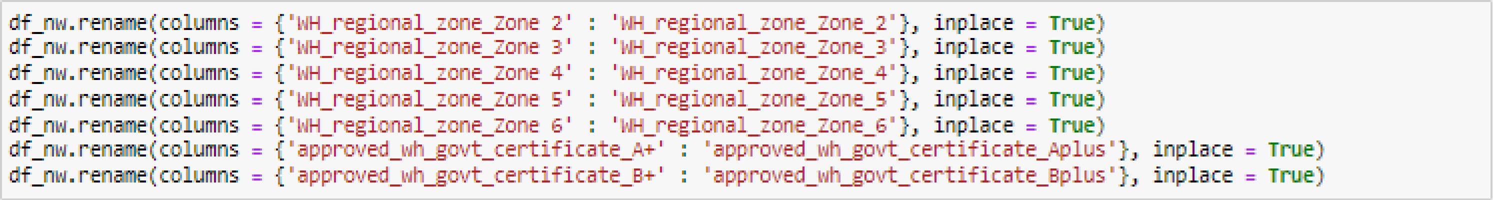

WH_capacity_size
50000
40000
30000
20000
10000
0
product_wg_ton
Zone 1
Zone 2
Zone 3
Zone 4
Zone 5
Zone 6
WH_regional_zone



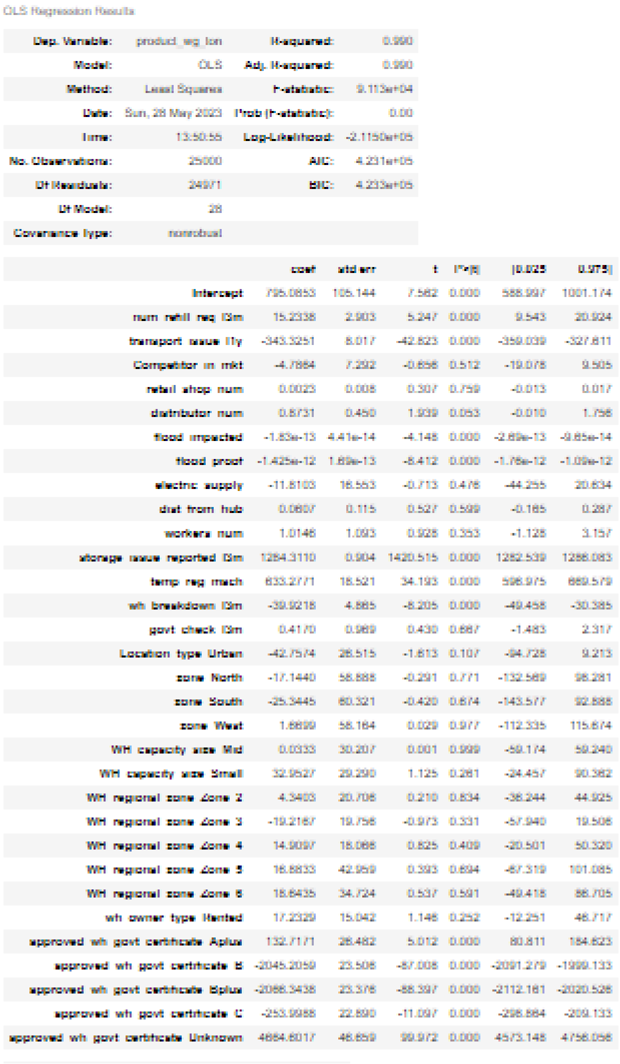




```
df_nw = pd.get_dummies(df, columns=['Location_type', 'zone', 'WH_capacity_size',
                                     'WH_regional_zone','wh_owner_type','approved_wh_govt_certificate'], drop_first=True)
```



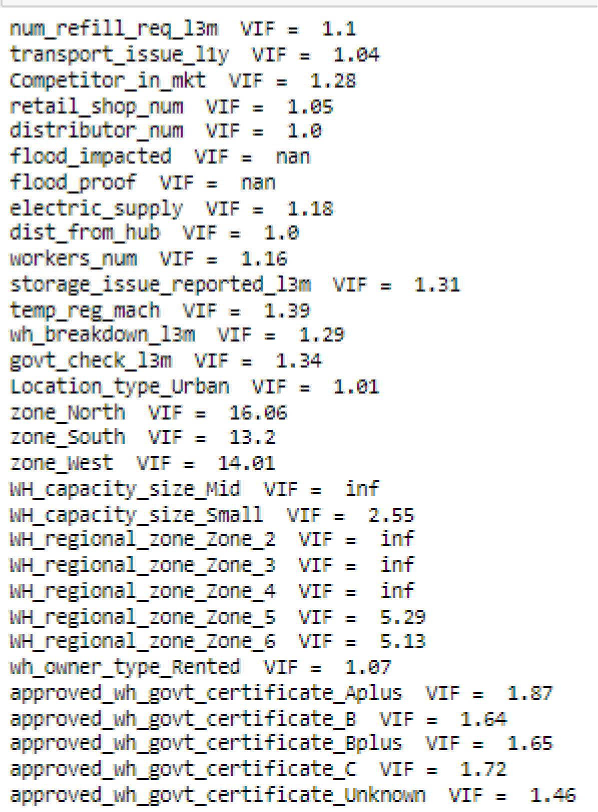
```
df_nw.rename(columns = {'WH_regional_zone_Zone 2' : 'WH_regional_zone_Zone_2'}, inplace = True)
df_nw.rename(columns = {'WH_regional_zone_Zone 3' : 'WH_regional_zone_Zone_3'}, inplace = True)
df_nw.rename(columns = {'WH_regional_zone_Zone 4' : 'WH_regional_zone_Zone_4'}, inplace = True)
df_nw.rename(columns = {'WH_regional_zone_Zone 5' : 'WH_regional_zone_Zone_5'}, inplace = True)
df_nw.rename(columns = {'WH_regional_zone_Zone 6' : 'WH_regional_zone_Zone_6'}, inplace = True)
df_nw.rename(columns = {'approved_wh_govt_certificate_A+' : 'approved_wh_govt_certificate_Aplus'}, inplace = True)
df_nw.rename(columns = {'approved_wh_govt_certificate_B+' : 'approved_wh_govt_certificate_Bplus'}, inplace = True)
```



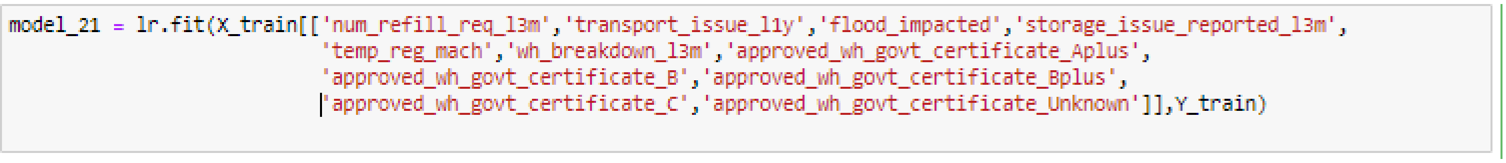
OLS Regression Results

Dep. Variable:	product wg ton	R-squared:	0.990
Model:	OLS	Adj. R-squared:	0.990
Method:	Least Squares	F-statistic:	9.113e+04
Date:	Sun, 28 May 2023	Prob (F-statistic):	0.00
Time:	13:50:55	Log-Likelihood:	-2.1150e+05
No. Observations:	25000	AIC:	4.231e+05
Df Residuals:	24971	BIC:	4.233e+05
Df Model:	28		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	795.0853	105.144	7.562	0.000	588.997	1001.174
num refill req l3m	15.2338	2.903	5.247	0.000	9.543	20.924
transport issue l1y	-343.3251	8.017	-42.823	0.000	-359.039	-327.611
Competitor in mkt	-4.7864	7.292	-0.656	0.512	-19.078	9.505
retail shop num	0.0023	0.008	0.307	0.759	-0.013	0.017
distributor num	0.8731	0.450	1.939	0.053	-0.010	1.756
flood impacted	-1.83e-13	4.41e-14	-4.148	0.000	-2.69e-13	-9.65e-14
flood proof	-1.425e-12	1.69e-13	-8.412	0.000	-1.76e-12	-1.09e-12
electric supply	-11.8103	16.563	-0.713	0.476	-44.255	20.634
dist from hub	0.0607	0.115	0.527	0.599	-0.165	0.287
workers num	1.0146	1.093	0.928	0.353	-1.128	3.157
storage issue reported l3m	1284.3110	0.904	1420.515	0.000	1282.539	1286.083
temp reg mach	633.2771	18.521	34.193	0.000	596.975	669.579
wh breakdown l3m	-39.9218	4.865	-8.205	0.000	-49.458	-30.385
govt check l3m	0.4170	0.969	0.430	0.667	-1.483	2.317
Location type Urban	-42.7574	26.515	-1.613	0.107	-94.728	9.213
zone North	-17.1440	58.888	-0.291	0.771	-132.569	98.281
zone South	-25.3445	60.321	-0.420	0.674	-143.577	92.888
zone West	1.6699	58.164	0.029	0.977	-112.335	115.674
WH capacity size Mid	0.0333	30.207	0.001	0.999	-59.174	59.240
WH capacity size Small	32.9527	29.290	1.125	0.261	-24.457	90.362
WH regional zone Zone 2	4.3403	20.706	0.210	0.834	-36.244	44.925
WH regional zone Zone 3	-19.2167	19.756	-0.973	0.331	-57.940	19.506
WH regional zone Zone 4	14.9097	18.066	0.825	0.409	-20.501	50.320
WH regional zone Zone 5	16.8833	42.959	0.393	0.694	-67.319	101.085
WH regional zone Zone 6	18.6435	34.724	0.537	0.591	-49.418	86.705
wh owner type Rented	17.2329	15.042	1.146	0.252	-12.251	46.717
approved wh govt certificate Aplus	132.7171	26.482	5.012	0.000	80.811	184.623
approved wh govt certificate B	-2045.2059	23.506	-87.008	0.000	-2091.279	-1999.133
approved wh govt certificate Bplus	-2066.3438	23.376	-88.397	0.000	-2112.161	-2020.526
approved wh govt certificate C	-253.9988	22.890	-11.097	0.000	-298.864	-209.133
approved wh govt certificate Unknown	4664.6017	46.659	99.972	0.000	4573.148	4756.056

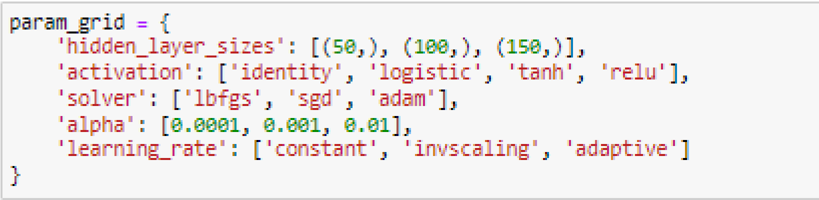


```
num_refill_req_l3m  VIF =  1.1
transport_issue_l1y  VIF =  1.04
Competitor_in_mkt  VIF =  1.28
retail_shop_num  VIF =  1.05
distributor_num  VIF =  1.0
flood_impacted  VIF =  nan
flood_proof  VIF =  nan
electric_supply  VIF =  1.18
dist_from_hub  VIF =  1.0
workers_num  VIF =  1.16
storage_issue_reported_l3m  VIF =  1.31
temp_reg_mach  VIF =  1.39
wh_breakdown_l3m  VIF =  1.29
govt_check_l3m  VIF =  1.34
Location_type_Urban  VIF =  1.01
zone_North  VIF =  16.06
zone_South  VIF =  13.2
zone_West  VIF =  14.01
WH_capacity_size_Mid  VIF =  inf
WH_capacity_size_Small  VIF =  2.55
WH_regional_zone_Zone_2  VIF =  inf
WH_regional_zone_Zone_3  VIF =  inf
WH_regional_zone_Zone_4  VIF =  inf
WH_regional_zone_Zone_5  VIF =  5.29
WH_regional_zone_Zone_6  VIF =  5.13
wh_owner_type_Rented  VIF =  1.07
approved_wh_govt_certificate_Aplus  VIF =  1.87
approved_wh_govt_certificate_B  VIF =  1.64
approved_wh_govt_certificate_Bplus  VIF =  1.65
approved_wh_govt_certificate_C  VIF =  1.72
approved_wh_govt_certificate_Unknown  VIF =  1.46
```

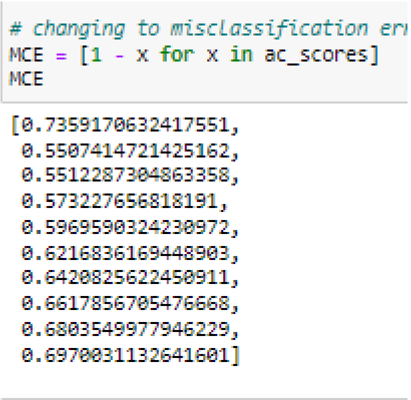


```
model_21 = lr.fit(X_train[['num_refill_req_l3m','transport_issue_l1y','flood_impacted','storage_issue_reported_l3m',
                           'temp_reg_mach','wh_breakdown_l3m','approved_wh_govt_certificate_Aplus',
                           'approved_wh_govt_certificate_B','approved_wh_govt_certificate_Bplus',
                           'approved_wh_govt_certificate_C','approved_wh_govt_certificate_Unknown']],Y_train)
```

Code for final model in linear regression.

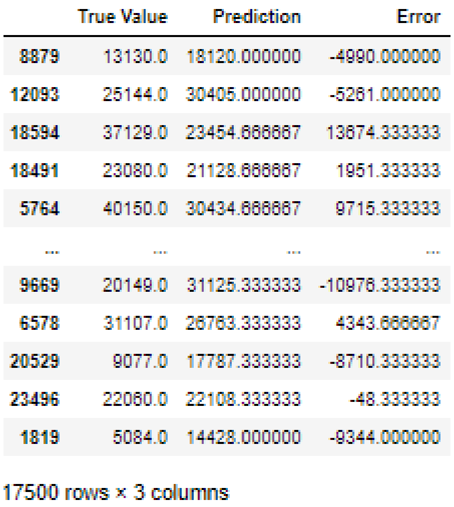


```
param_grid = {
    'hidden_layer_sizes': [(50,), (100,), (150,)],
    'activation': ['identity', 'logistic', 'tanh', 'relu'],
    'solver': ['lbfgs', 'sgd', 'adam'],
    'alpha': [0.0001, 0.001, 0.01],
    'learning_rate': ['constant', 'invscaling', 'adaptive']
}
```



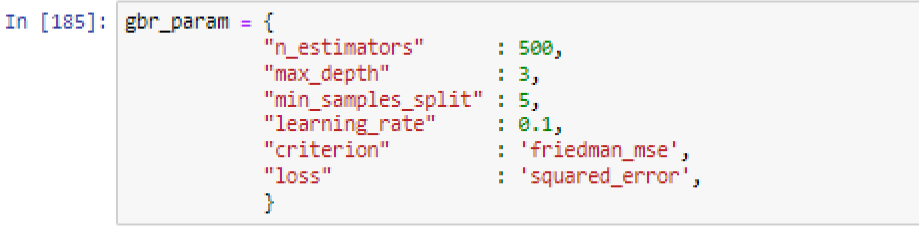
```
# changing to misclassification er
MCE = [1 - x for x in ac_scores]
MCE
```

```
[0.7359170632417551,
 0.5507414721425162,
 0.5512287304863358,
 0.573227656818191,
 0.5969590324230972,
 0.621683616944890З,
 0.6420825622450911,
 0.6617856705476668,
 0.6803549977946229,
 0.6970031132641601]
```



	True Value	Prediction	Error
8879	13130.0	18120.000000	-4990.000000
12093	25144.0	30405.000000	-5261.000000
18594	37129.0	23454.666667	13674.333333
18491	23080.0	21128.666667	1951.333333
5764	40150.0	30434.666667	9715.333333
...	...	...	...
9669	20149.0	31125.333333	-10976.333333
6578	31107.0	26763.333333	4343.666667
20529	9077.0	17787.333333	-8710.333333
23496	22060.0	22108.333333	-48.333333
1819	5084.0	14428.000000	-9344.000000

17500 rows × 3 columns



```
In [185]: gbr_param = {
                       "n_estimators"      : 500,
                       "max_depth"         : 3,
                       "min_samples_split" : 5,
                       "learning_rate"     : 0.1,
                       "criterion"         : 'friedman_mse',
                       "loss"              : 'squared_error',
                       }
```