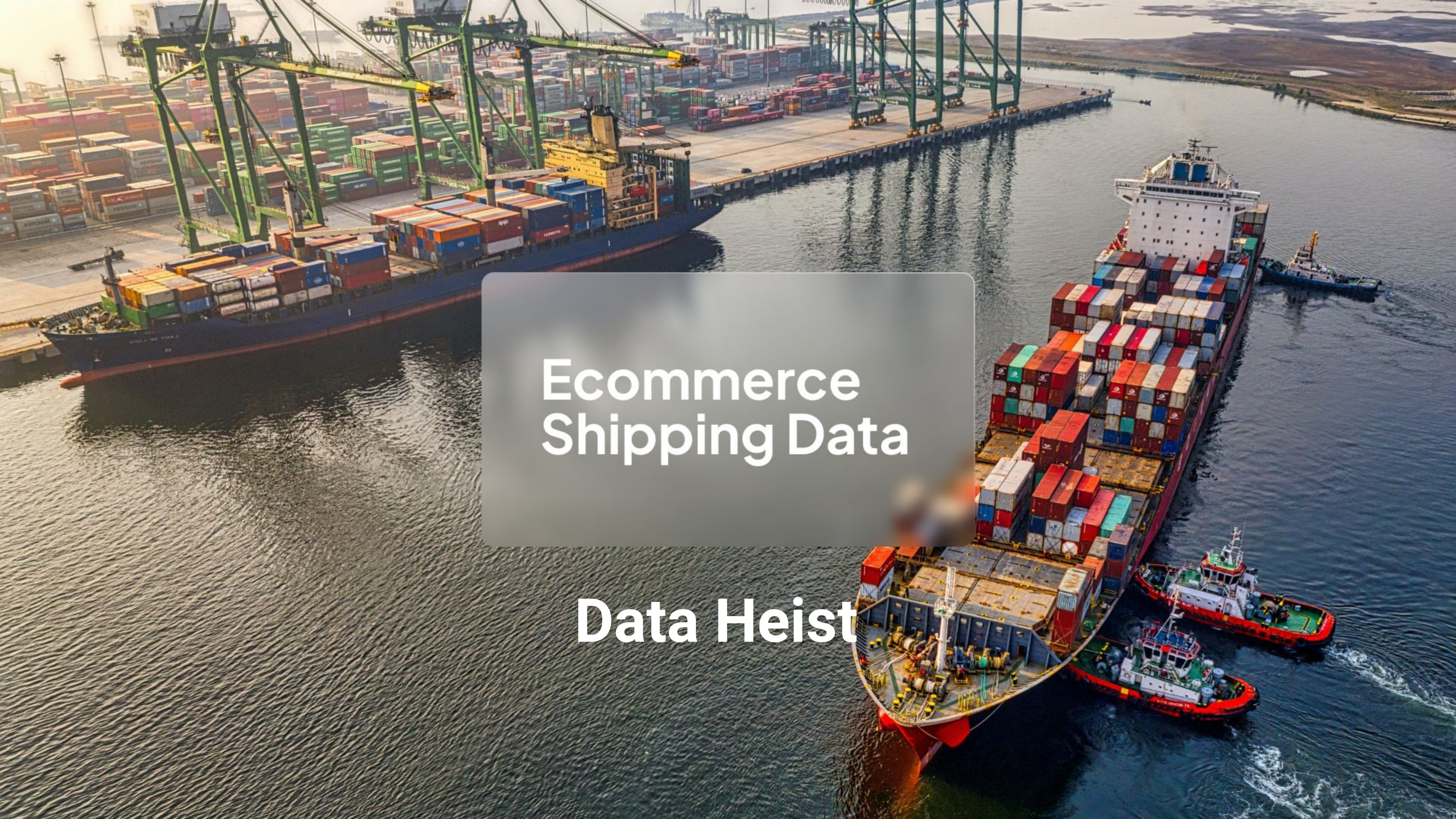

Ecommerce
Shipping Data
Data Heist

MEET OUR TEAM
Data Heist

Elvis

Luthfi

Fuji

Haolia

Cyntia

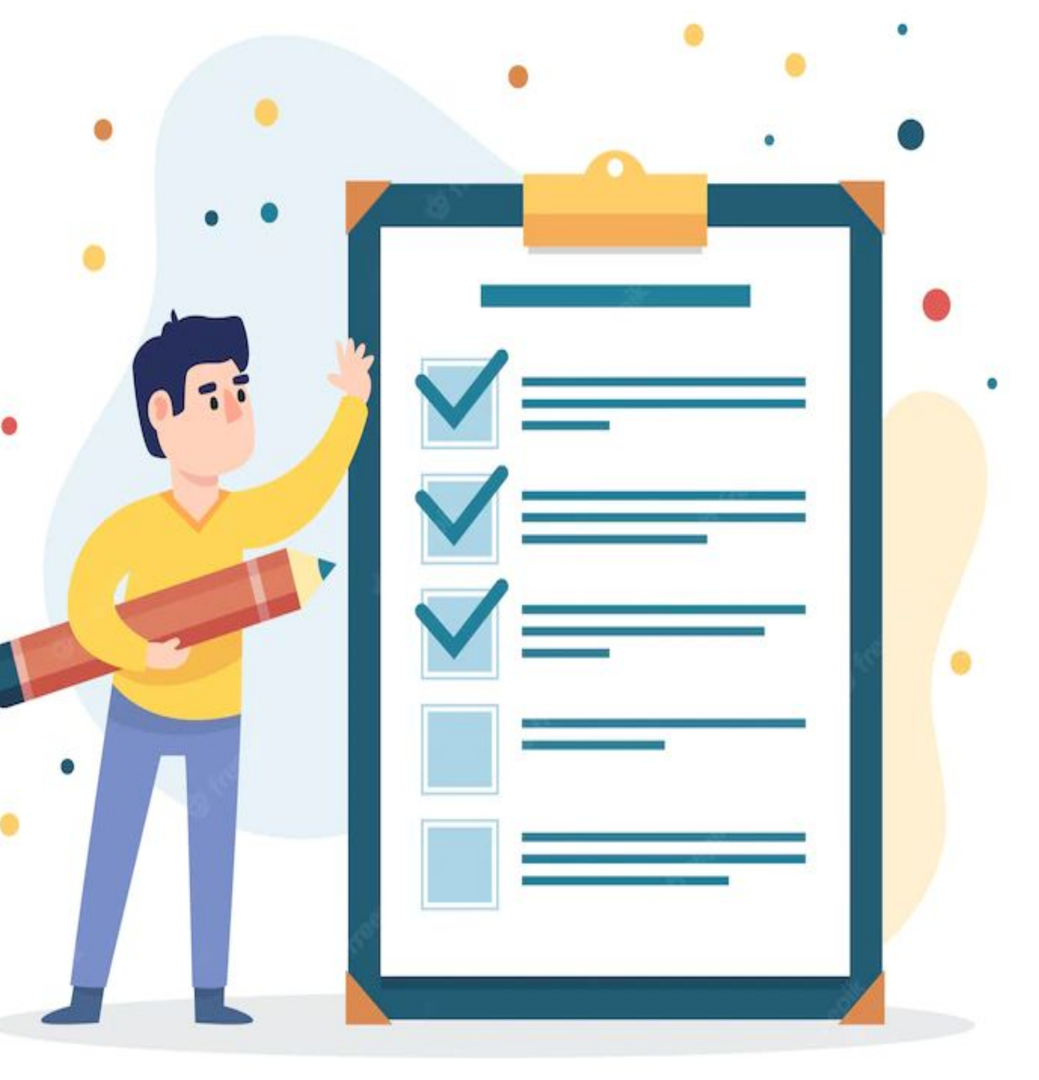


Table of Contents

Research Background

WHO ARE WE ?

DATA HEIST

Data Scientist from a consulting company engaged in business consulting

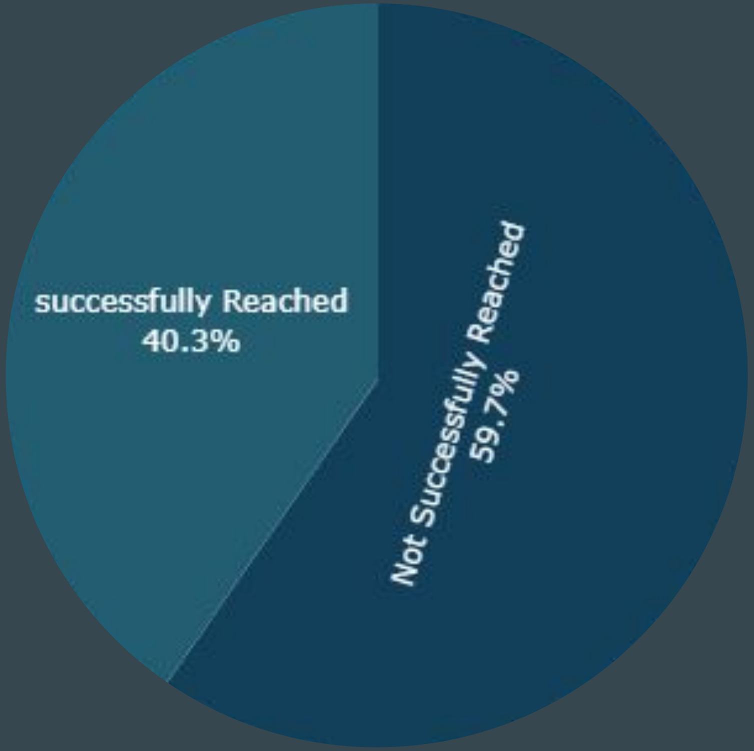
AMAJON

International e-commerce focused on selling electronic products

E-commerce (electronic commerce) is the buying and selling of goods and services, or the transmitting of funds or data, over an electronic network, to reduce service-costs aimed at improving product quality and delivery quality.

Problem Statement

- There was a late delivery of 59.7%
- Customer Rating 2.99



Solution :

We as consultants can help companies to increase customer retention which at the same time is able to increase the profits of our client companies.

GOAL : Increase the percentage of delivery on time and increase customer rating

OBJECTIVE : Provide insight and build machine learning models to predict Delivery On time

BUSINESS METRIC : *Delivery on time, customer rating*

DataSet
&
EDA

About this Dataset

ID

Customer_care_calls

Customer_Rating

Cost_of_the_product

Prior_purchases

Discount_offered

Weight_in_gms

Warehouse_block

Mode_of_shipment

Product_importance

Gender

Reach.on.time_Y.N

NO

Duplicate data or missing data

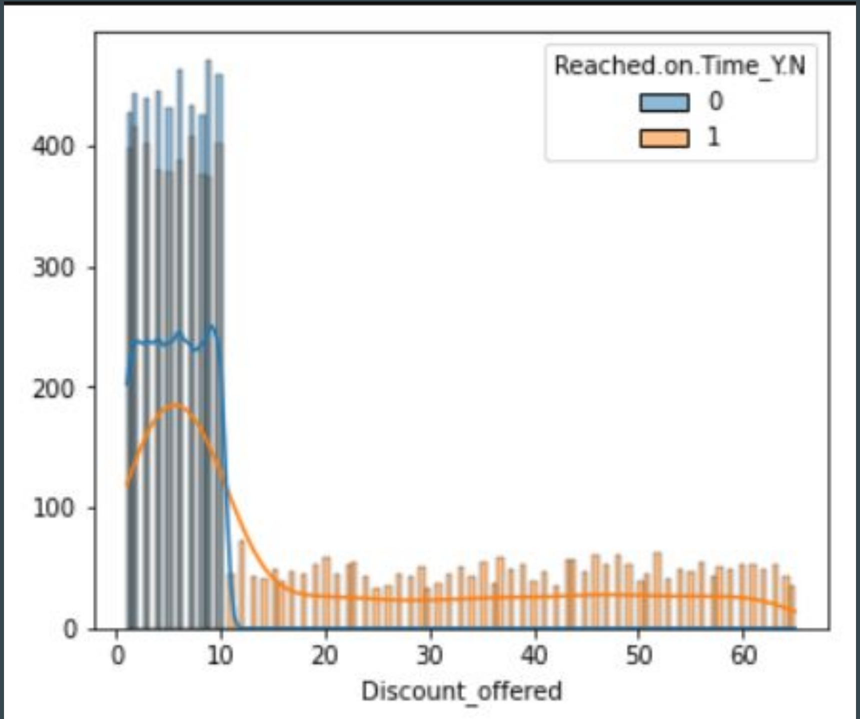
Strange data type

- Consist of 10999 rows and 12 column
- Column"ID" will be drop

Discount_Offered

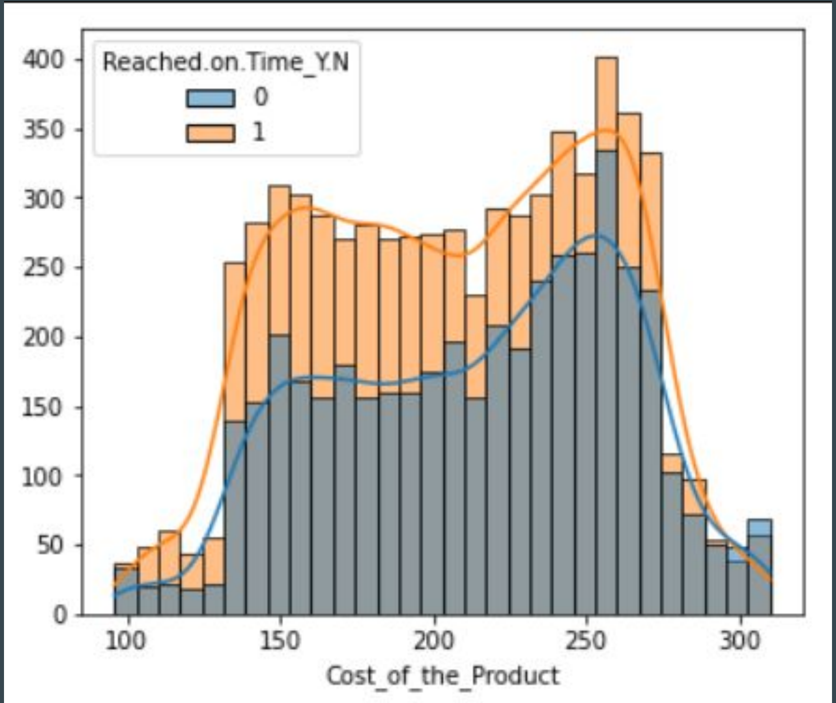
When discount < $10, delivery on time is great

When discount > $10, delivery on time is low



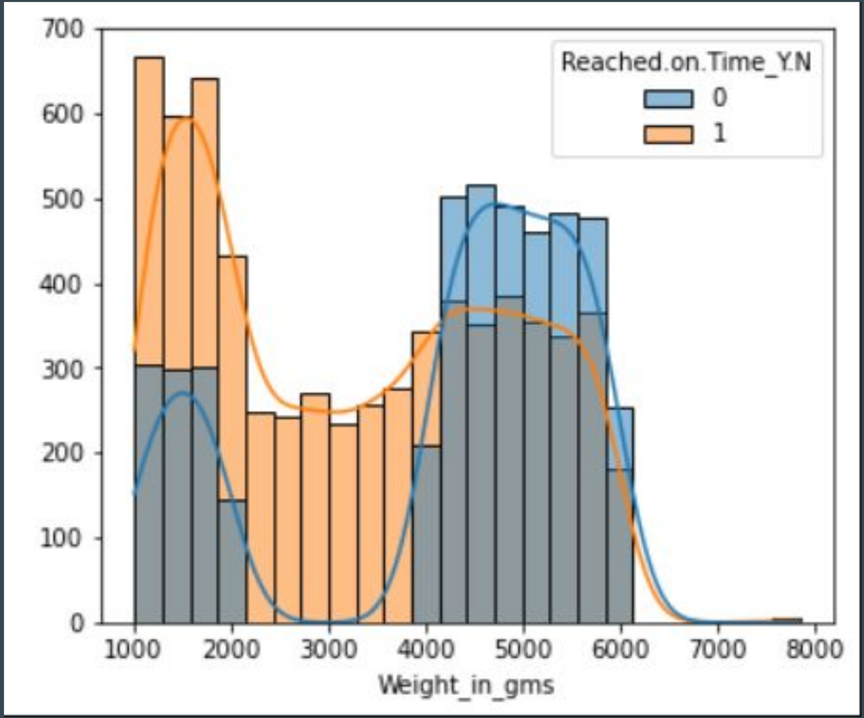
Cost_of_the_Product

Late delivery increased by $130 and decreased by $275

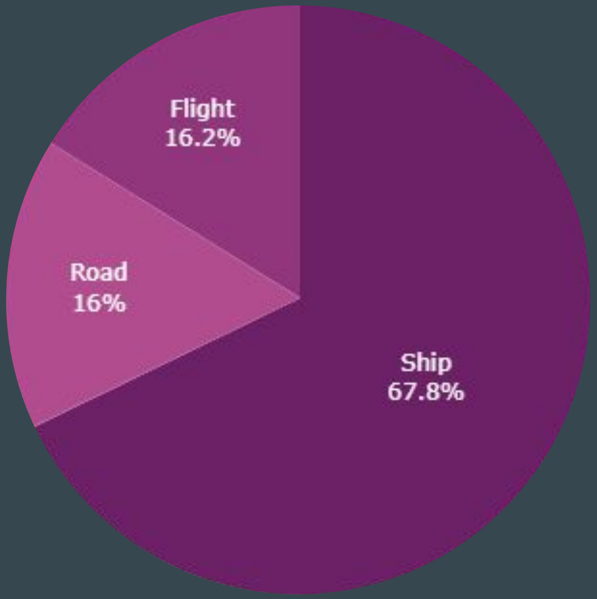


Weight_in_gms

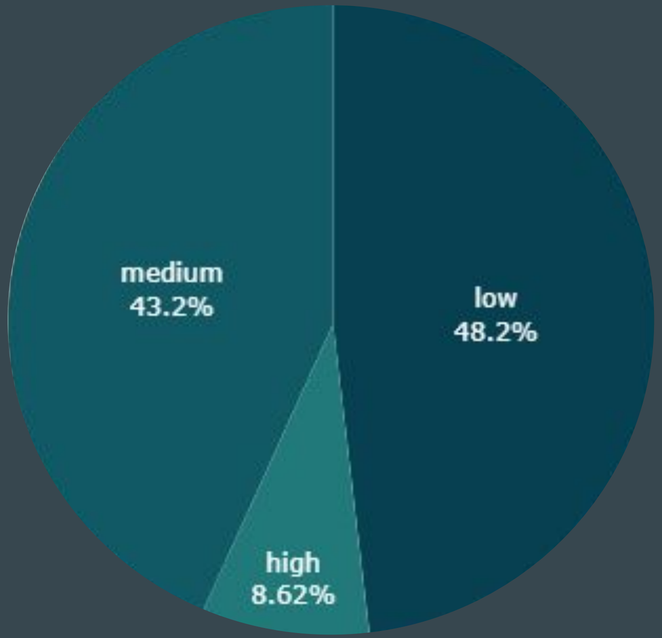
Late delivery is dominant on product weight below 2 kg



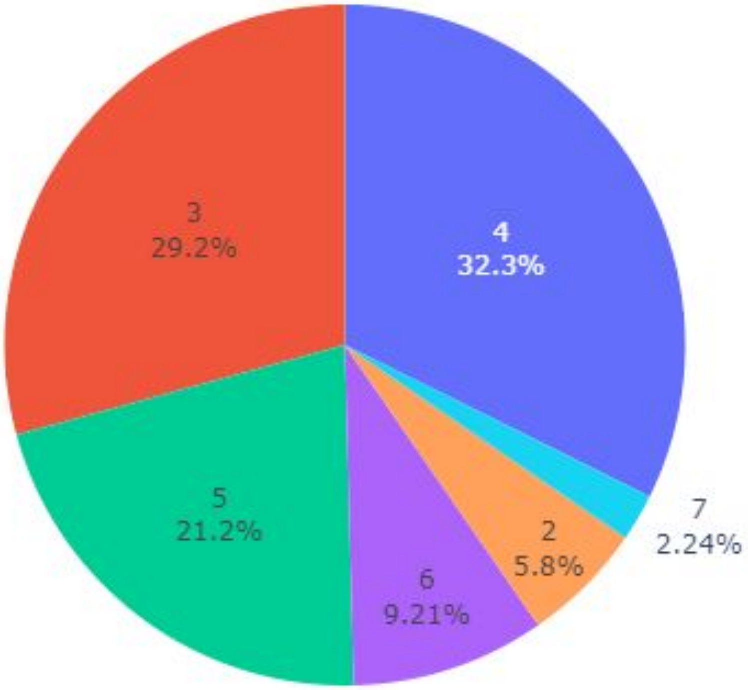
Mode_of_shipment



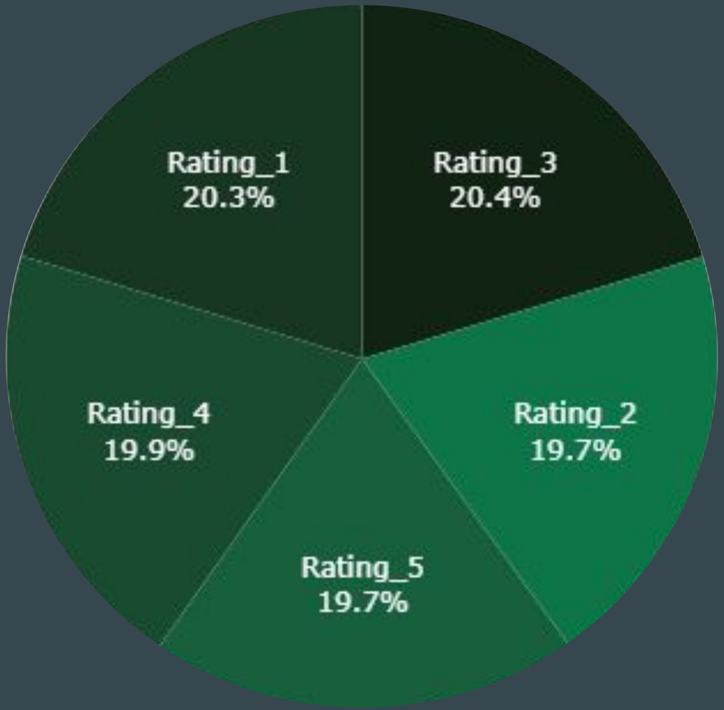
Product_Importance



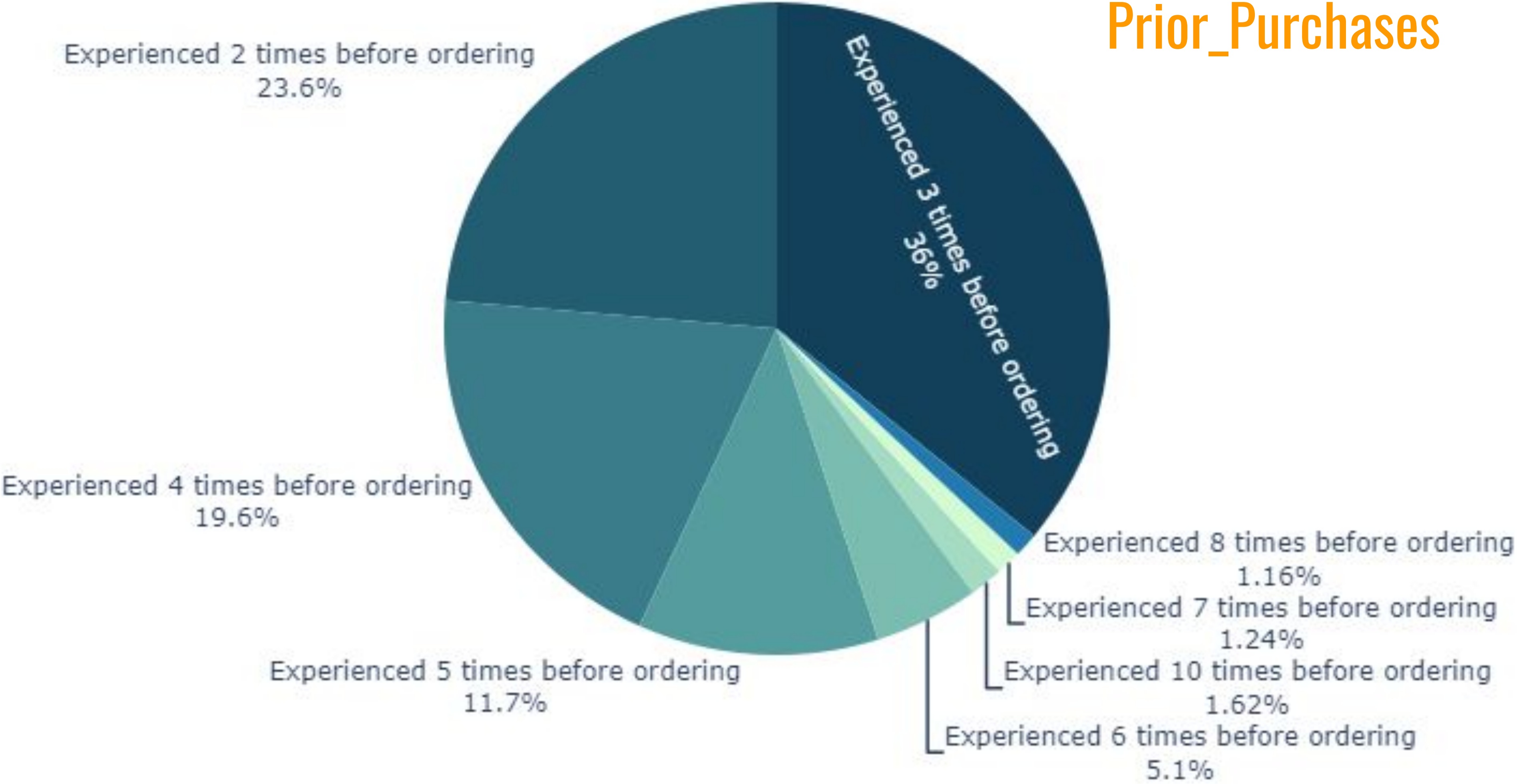
Customer_Care_Calls



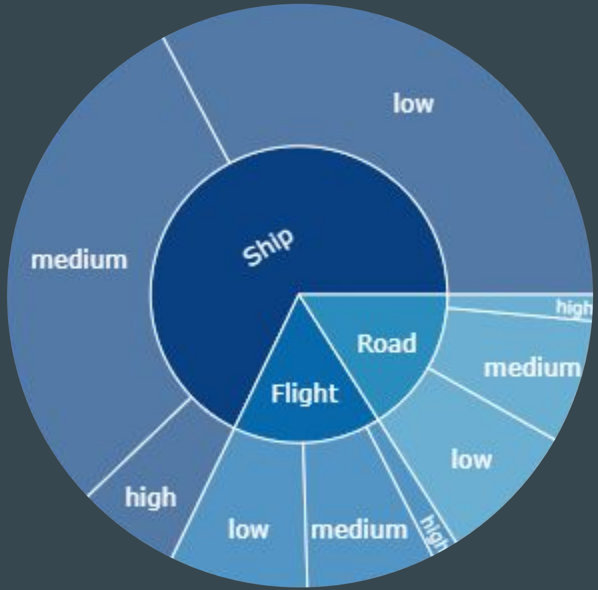
Ratings



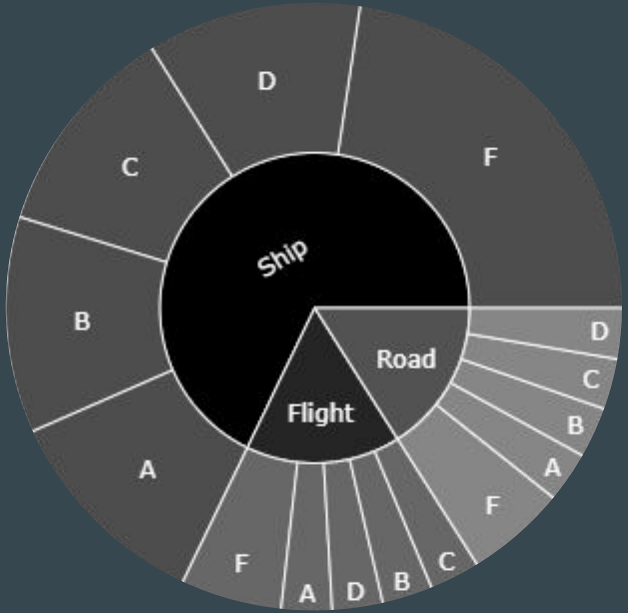
Prior_Purchases



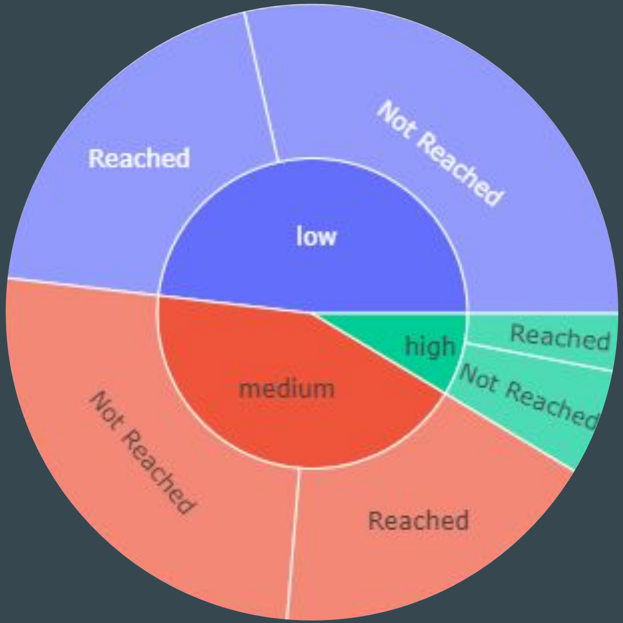
Mode_of_shipment belong to product importance



Mode_of_shipment belong to warehouse



Product_importance_with_reached



Pre-Processing Data

Split Data 80: 20

Handling Missing Value (No missing value)

Handling Outlier

- Outliers in the Prior_purchases column are not removed because their values are still within reasonable limits
- Outlier in the Discount_offered column are not removed because the discount given is still within reasonable limits

Boxcox Transformation

Scaling with standar scaler

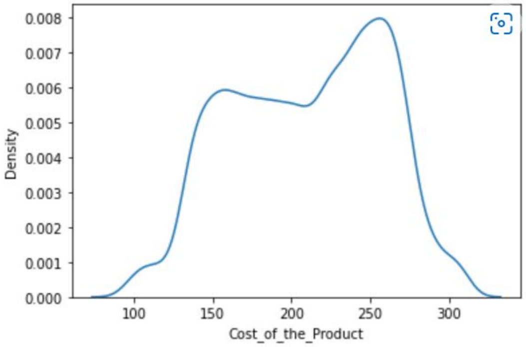
Feature Encoding Feature selection (ID)

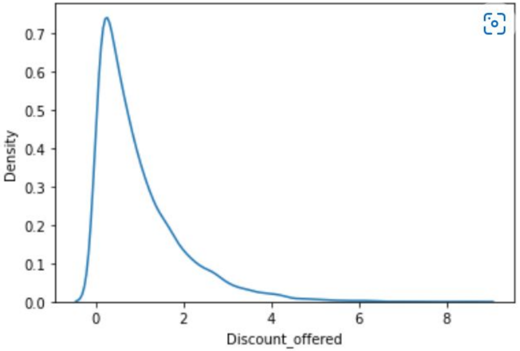
- One hot encoding (*Mode of shipment*)
- Label encoding *(Gender, Product importance, Warehouse block)*

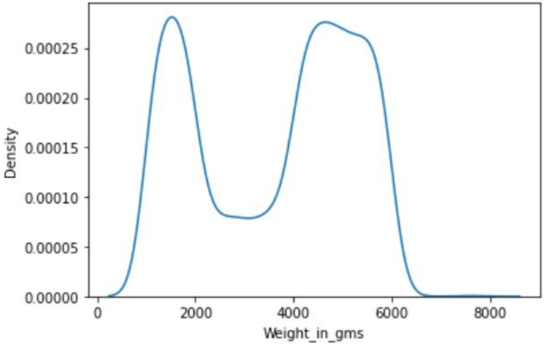
Class Imbalance

- Target feature 60:40 (balance)

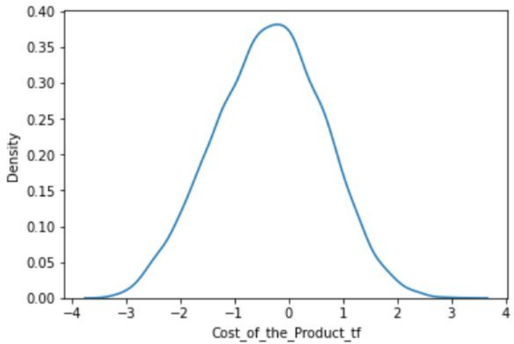
Before Transformation

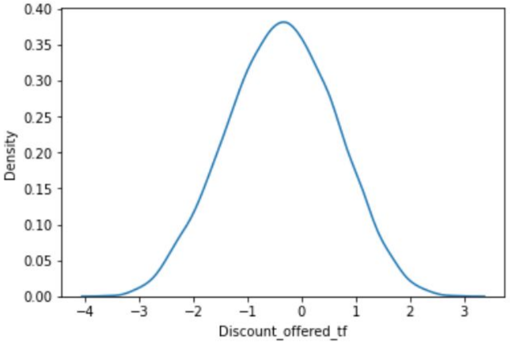


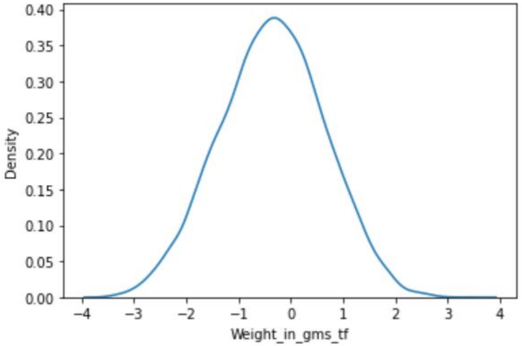




After Transformation







Model
&
Evaluation

Model Comparison After Hyperparameter Tuning

Model	Train			Test		
	Accuracy	Recall	Roc_Auc	Accuracy	Recall	Roc_Auc
Logistic Regression	0.632	0.779	0.599	0.780	0.630	0.594
Decision Tree	0.597	1.000	0.500	0.593	1.000	0.500
KNN	0.696	0.768	0.677	0.635	0.710	0.618
SVM	0.597	1.000	0.500	0.593	1.000	0.500
Random Forest	1.000	1.000	1.000	0.661	0.668	0.659
XGBoost	0.931	1.000	0.914	0.626	0.800	0.586
AdaBoost	0.912	0.864	0.924	0.668	0.628	0.677
CatBoost	0.781	0.727	0.794	0.663	0.629	0.671

Hyperparameter Tuning

- **KNN** has the best performance with **accuracy 0.635** and **ROC-AUC score 0.628**
- **71% Recall** of all late deliveries, **only 29% were not detected late** (False Negative) meaning that the percentage of late deliveries can be predicted better.

Predicted

Actual	Negative	Positif
Negative	(TN) 471	(FN) 378
Positif	(FP) 424	(TP) 927

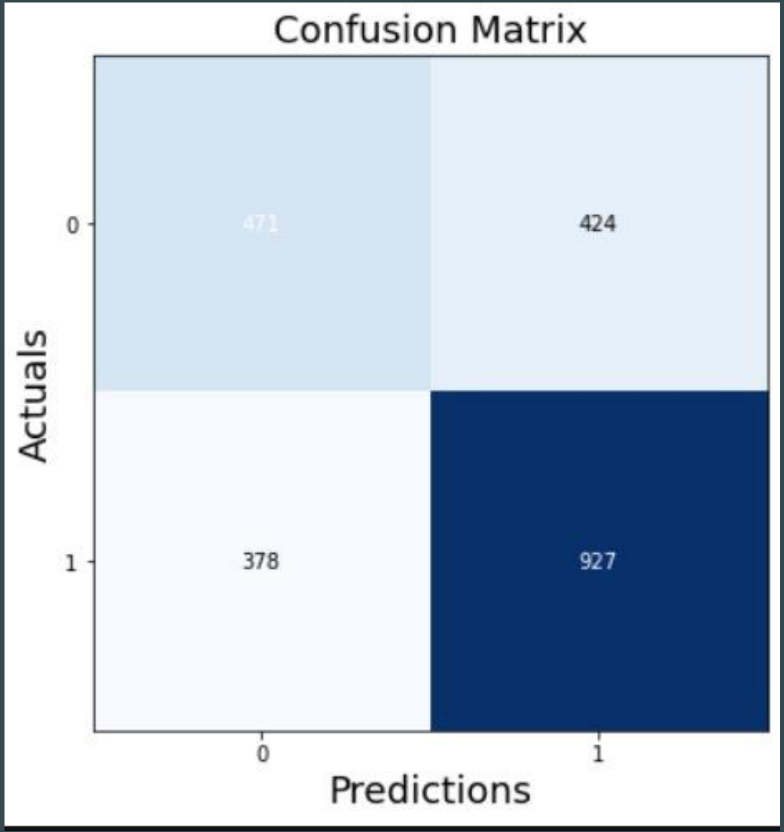
Model	Train			Test		
	Accuracy	Recall	Roc_Auc	Accuracy	Recall	Roc_Auc
KNN	0.696	0.768	0.677	0.635	0.710	0.618

Confusion Matrix

Accuracy = 0.635

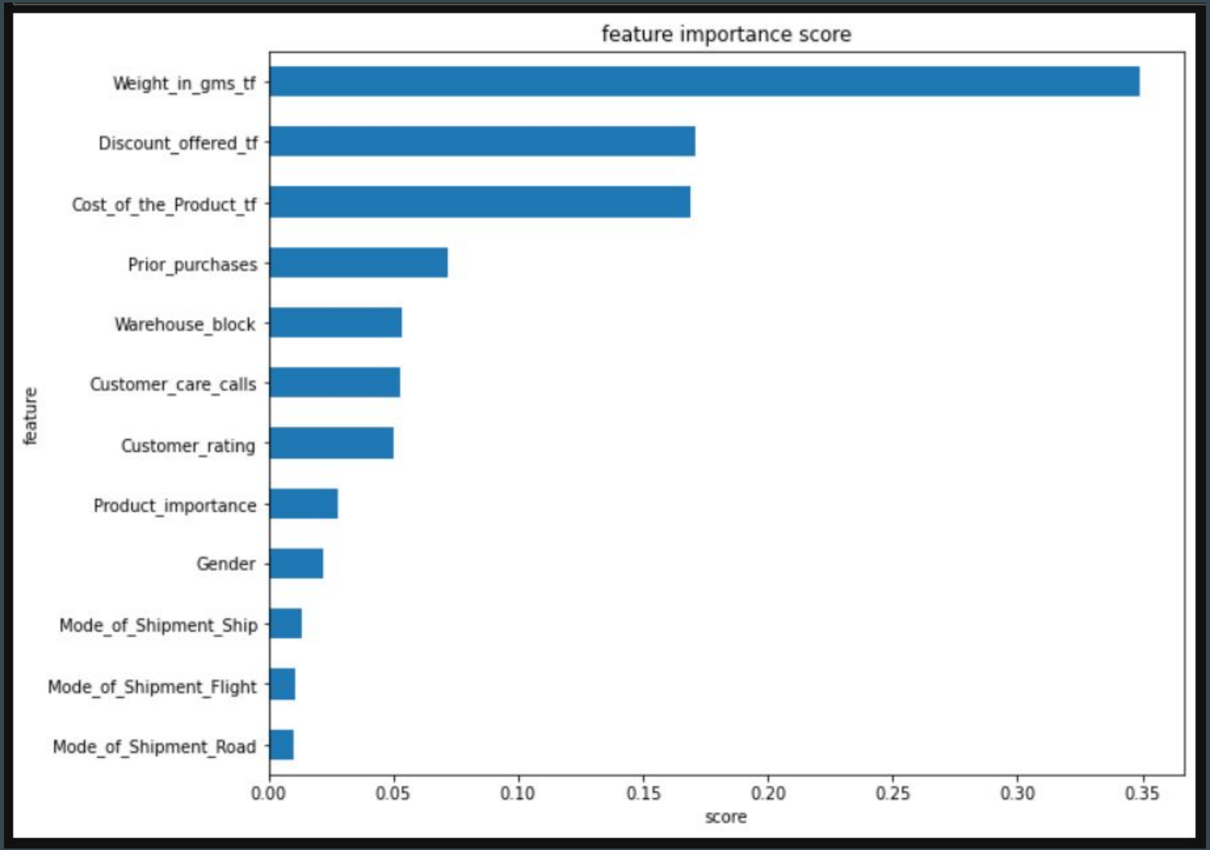
Precission = 0.686

Recall = 0.710



Feature Importance

1. Weight in Grams
2. Discount Offered
3. Cost of the Product
4. Prior Purchases



Business Insight & Recommendation

On Time Growth Calculation

EXISTING		
Feature	#	%
Delivery	10.999	100%
Late	6.563	59.7%
On Time	4.436	40.3%

AFTER MODEL PREDICTION			
Feature	var	#	%
Delivery	a	10.999	100%
Late	b	6.563	59.7%
Predicted Late	c	4.660	71%
Predicted on Time	d	1.903	29%
Late after Prediction	e = (b-c)	1.903	17.3%
On Time	f	4.436	40.3%
On Time after Prediction	g = (f+c)	9.096	82.70%
On Time Growth Rate	4.436 to 9.096 = 105%		

Potential Revenue Loss Saved Calculation

	Delivery a	Total Cost of the Product b	Total Discount c	Total Revenue d = (b-c)	Avg Revenue e = (d/a)
Delivery	10.999	$ 2.311.955	$147.092	$2.164.863	$196.8

<table>
<tr><th></th><th>Delivery
a</th><th>Average Revenue
b</th><th>Potential Revenue
c = (a * b)</th><th>%
b</th></tr>
<tr><td>Late</td><td>6.563</td><td rowspan="3">$196.8</td><td>$ 1.291.598,4</td><td>100%</td></tr>
<tr><td>Predicted on Time</td><td>1.903</td><td>$ 374.510,4</td><td>29%</td></tr>
<tr><td>Predicted Late</td><td>4.660</td><td>$ 917.088</td><td>71%</td></tr>
</table>

Rating Growth Calculation

	Delivery a	Total Rating b	Avg Rate c = (b / a)
Delivery	10.999	32.893	2.99
Predicted Late Customers potentially increased their rating by 1 (except if the customer already gave rating = 5)	4.660 - 932(20%) = 3.728	32.893 + 3.728 = 36.621	3.33
Rating Growth Rate	2.99 to 3.33 = 11.7%		

	Late	Rate 5
Customers	6.563	1.371 (20%)

20% from all Late customer give 5 rating

Business Insight

<table>
<tr><th>Before using the model</th><th>After correctly using the KNN model (Recall 0.71)</th></tr>
<tr><td>The number of customers who made transactions was 10999 (4436 received on time)</td><td rowspan="2">Orders on time based on predictions:
Late order prediction + Order On Time
4660 + 4436 = 9096 (82.70%)</td></tr>
<tr><td>Percentage of orders arriving on time
4463/10999 = 40.33%</td></tr>
</table>

Business Recommendation

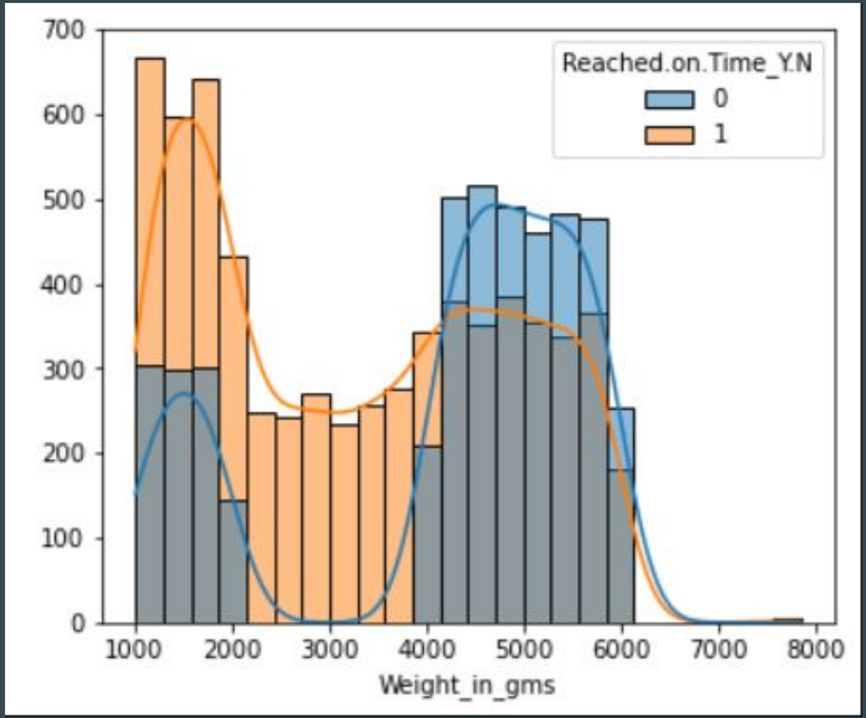
- Create a delivery duration in the form of a range
- Increase the estimated duration of package delivery to the customer as a prevention of delays.
- Referring to Prior purchases, we create member ranking programs such as golden, silver, and bronze so that discounting will be more effective so that we can retain customers as well.
- Added shipping tracking notifications so this can reduce customer care calls

Business Insight

- **Weight in grams**,

  we can categorize the priority of an item, so the shipping process can be adjusted to the mode of shipment.

  For example, goods with heavy categories can be sent using Ship mode and lighter goods can use other faster modes such as flight.



Business Recommendation

From these results, it can be concluded that giving a discount is quite influential on delivery on time, if the discount given is below 10$, the number of delivery on time will increase.

This is because if there is a big discount, the number of orders will increase and so will delivery late.

If a big discount is offered > more orders > delivery late retention increases

So it is very important for Amajon to think about the discount scheme that will be offered so that the delivery stays according to the estimated time.

Other strategies we can recommend:

- Add delivery distance data so that it can provide a clearer picture of the delivery of the goods.
- Adding other shipment modes such as two-wheeled land transportation and trains

THANK YOU

Evaluation Metrics

- Accuracy : Used because each label has the same importance/ balance
- Recall does not allow a large False Negative value so that the True Positive Rate can be identified properly.
- Roc AUC ensure that the model can distinguish classes well (not all data is predicted positive/negative) AUC perfect = 1, baseline = 0.5

Hyperparameter Tuning Notes

The hyperparameters used for tuning the best model are:

- **eta** : step size shrinkage to prevent overfitting
- **gamma** : minimum loss reduction needed to create the next partition
- **max_depth** : maximum depth of tree
- **min_Chid wight** : the minimum number of weights on a "child" (partition) where the higher this parameter, the more conservative the model
- **colsample_bytree** : subsample ratio in tree construction
- **lambda** : L2 regularization, where the higher the parameter, the more conservative the model
- **alpha** :L2 regularization, where the higher the parameter, the more conservative the model
- **tree_method** : tree construction algorithm

1- n_neighbors

Since it’s a very fundamental parameter in kNN algorithm, n_neighbors can directly affect the accuracy of the results as well as runtime performance.

2- algorithm

This parameter defines the algorithm that’s used to calculate neighbor distances. It is “auto” by default in Scikit-Learn which works pretty well to identify the ideal algorithm that should be used to calculate distances between sample points.

3- leaf_size

leaf_size offers a great opportunity to fine tune kNN algorithm when performance is a critical criteria.

4- Distance metrics in kNN

metric parameter is used to define the method for distance calculations between sample points in kNN algorithm. By default metric is minkowski with parameter 2 (p=2).