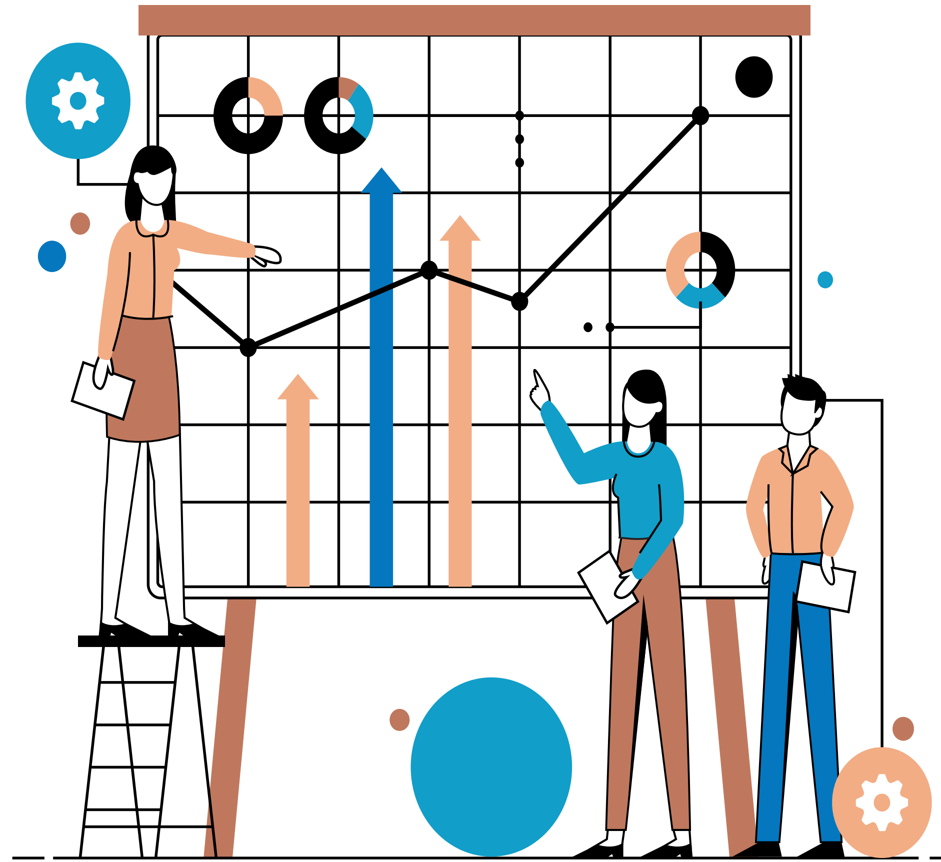
# E-Commerce Shipping Classification Modelling

W3.Solutions()



# Table of Contents

# W3.Solutions()

Data Consultant

An international e-commerce company that sell electronic product call W3.Solutions() to discover key insights & studies from their customer database


M. Hamzah

M. Alfian Prasetyo

Mohammed Abyannash

Gesta Putra G.

Fajar Arif K.

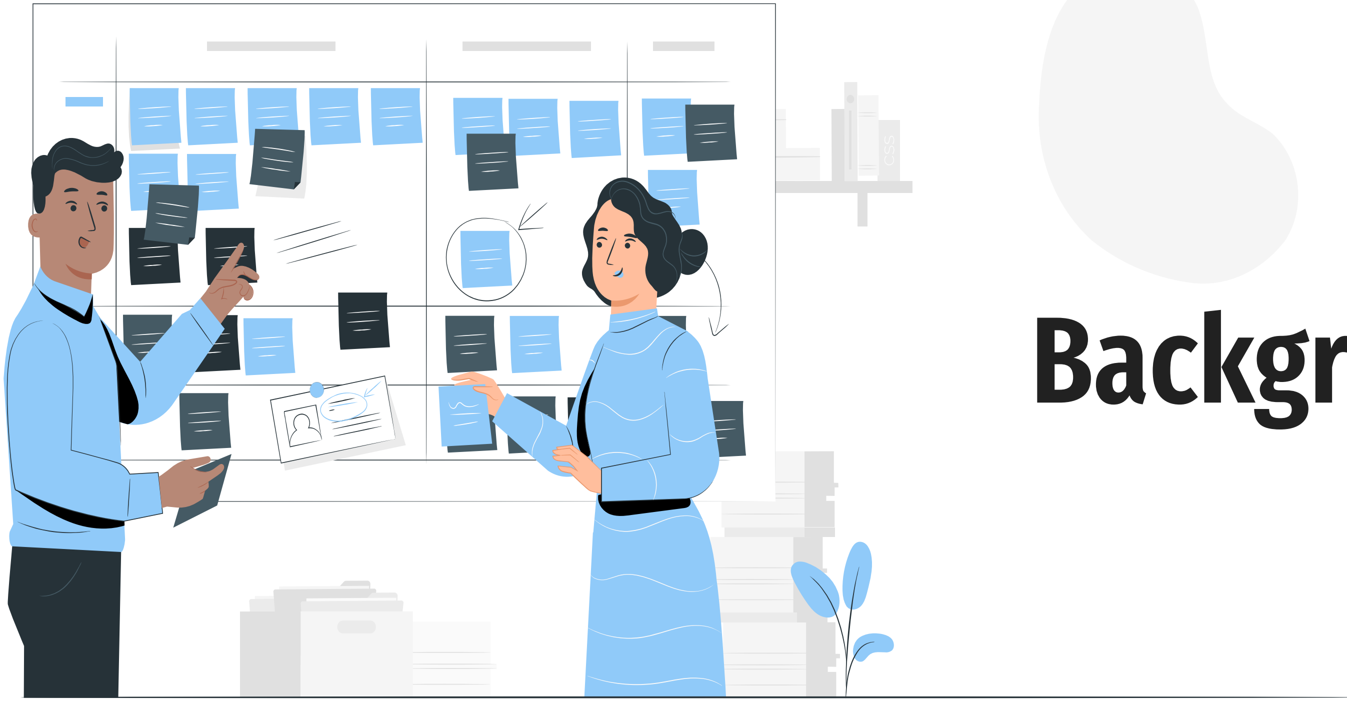

01

02

03

04

05

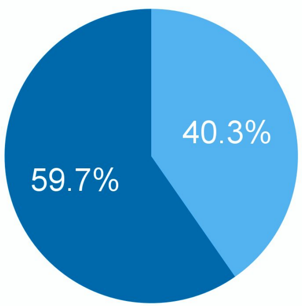


# 01

# Background

# BACKGROUND PROBLEM

Source : MHL News & Last Mile
Convey's 2018 Last-Mile Delivery Report




**59.7%** of the Business E-Commerce Deliveries Are **Late**

**6563 of 10999 Customers**

**01 LATE**

**87%** Online shoppers identified **shipping speed** as a **key factor** for online shoppers to shop aga

**02 DISSATISFACTION**

In face, price is not even as important as speed since **67%** online shoppers **would pay more** to get same day delivery

**03 STOP SHOPPING**

**84%** online shoppers are **unlikely to return** after a poor delivery experience.

**55%** online shoppers will **stop shopping** after receiving late delivery twice

**04 REVENUE LOSS**

Potential profits **will lose** because the customer left.

**52%** online shoppers expect a **refund** or discount on shipping cost after receiving late delivery

# BACKRGOUND PROBLEM

W3.Solutions() as a data consultant will analyze insight & make predictions model about whether the delivery will be received late/on time by the customer to help solve e-commerce shipping problem

# BACKGROUND PROBLEM

| Current Condition |
| --- |
| Most of the E-Commerce Deliveries are not reached on time |

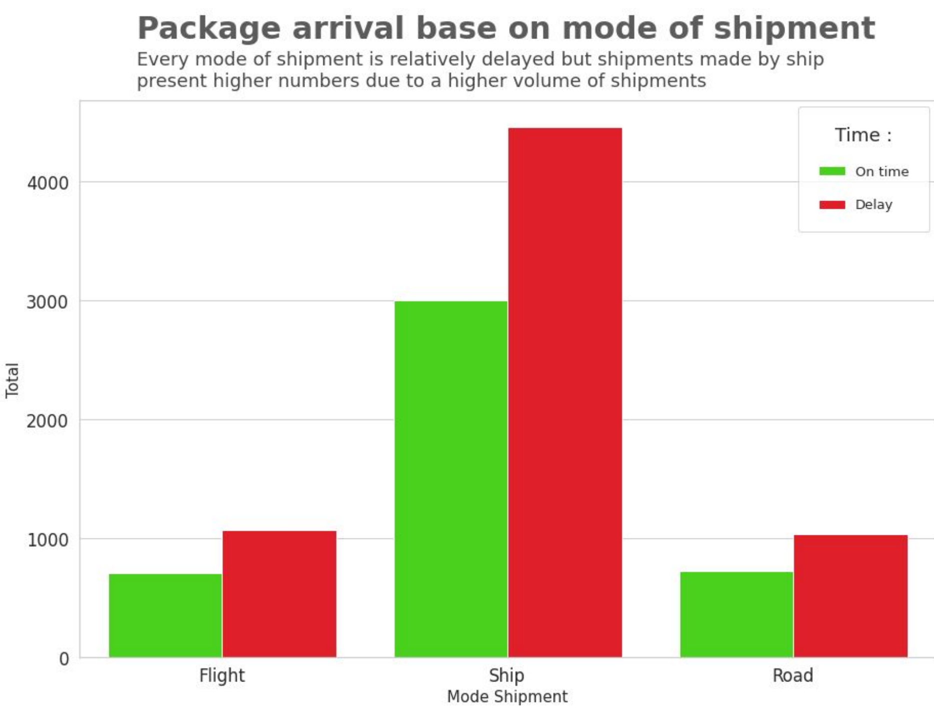
| Machine Learning Approach | | |
| --- | --- | --- |
| Insight | Action | Impact |
| Finding Pattern from database feature | Predictive model | Insight & Recommendations |

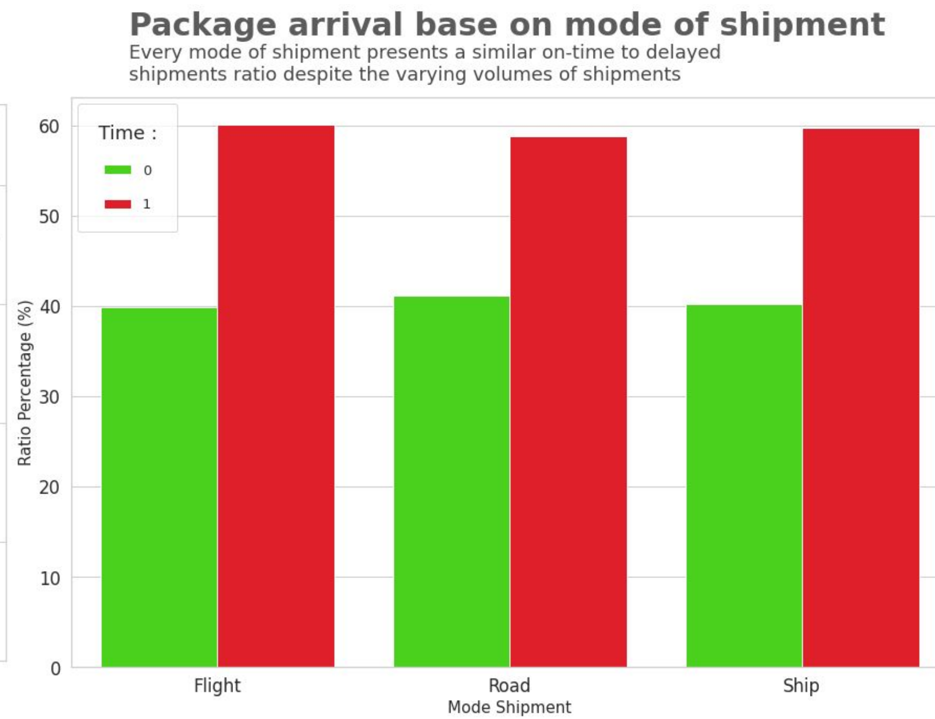
| Business Approach | |
| --- | --- |
| Action | Business Impact |
| Recommendation analysis & decision | On time rate, customer satisfaction & safe potential revenue loss |

# 02

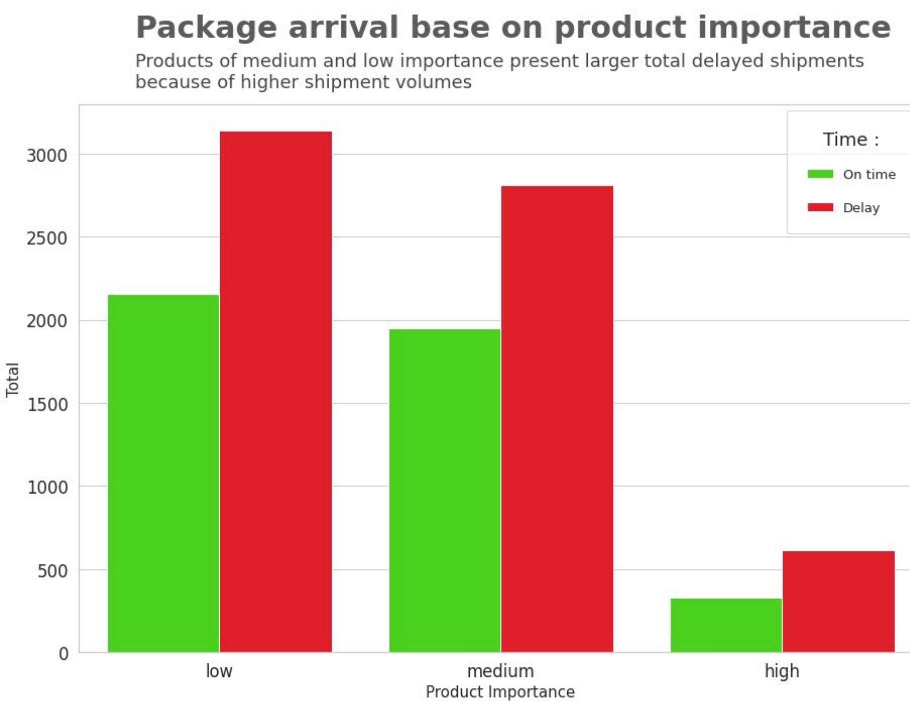
# The Insight

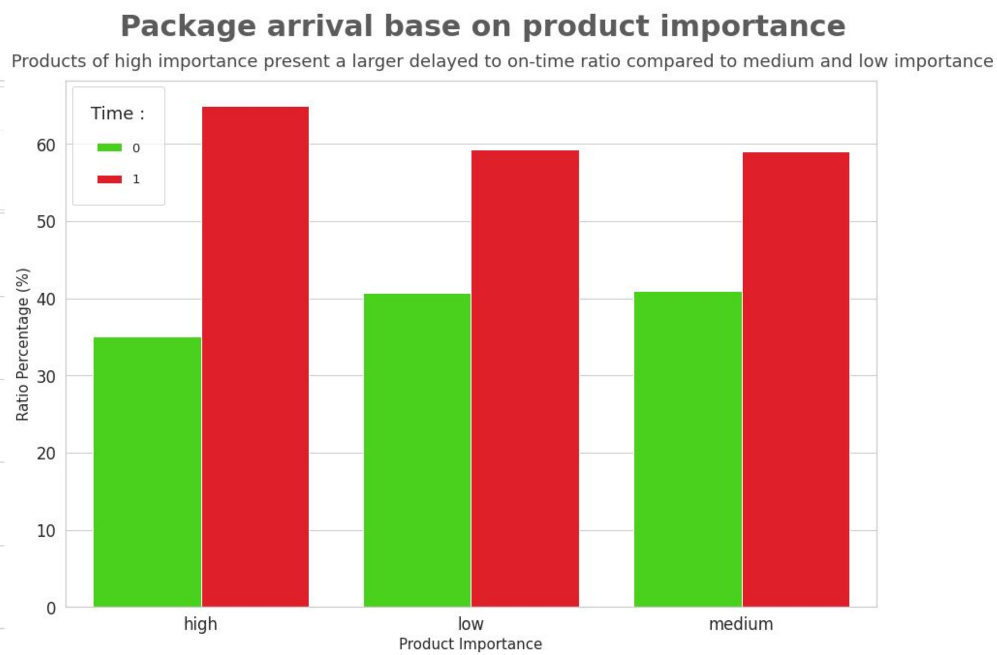
# Insight Mode Of Shipment



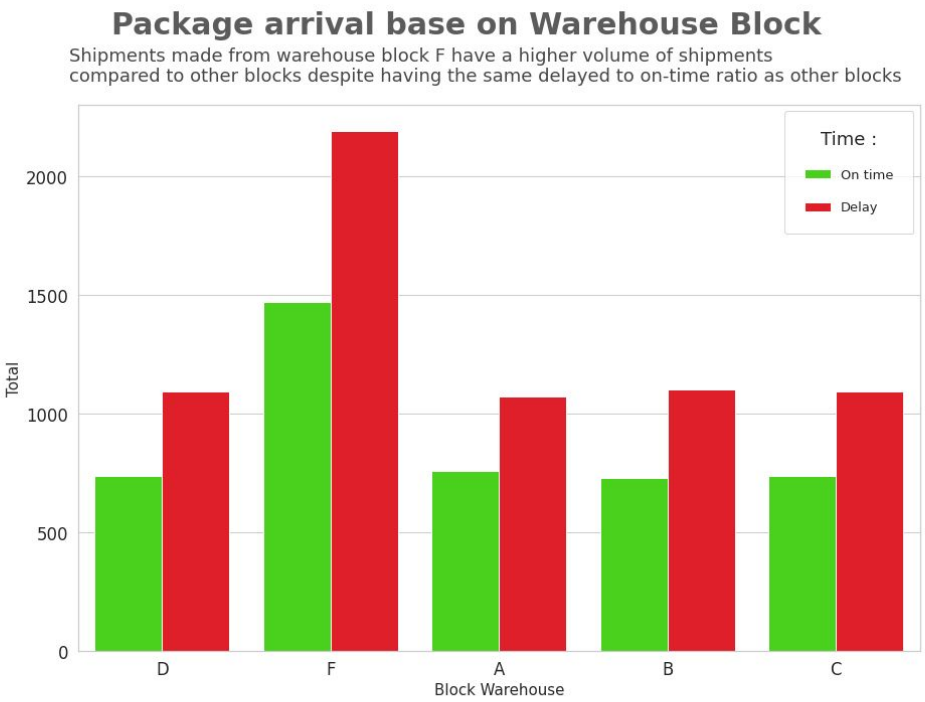



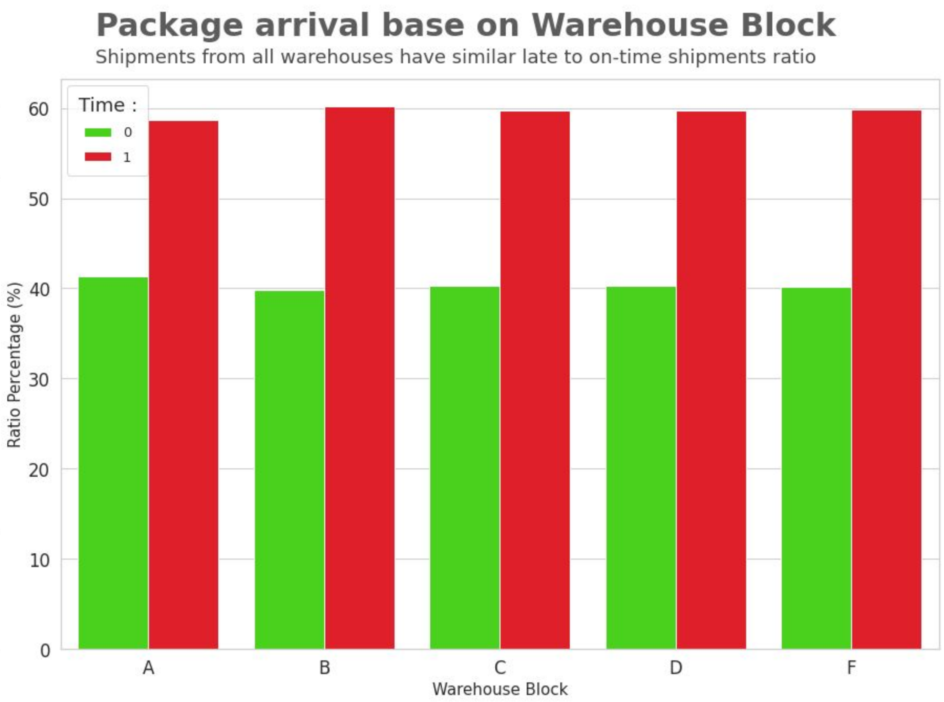
# Insight Product Importance



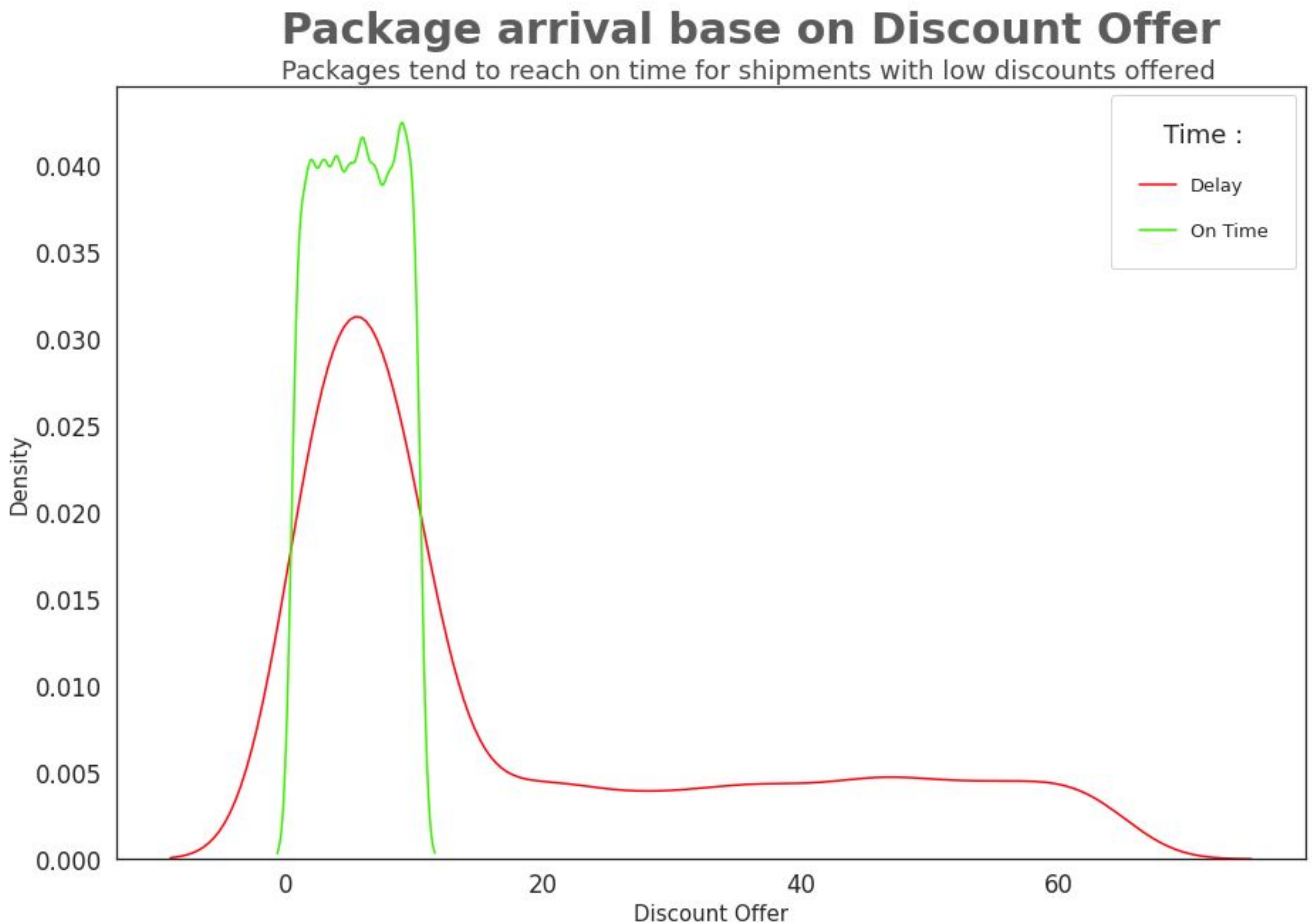



# Insight Warehouse Block



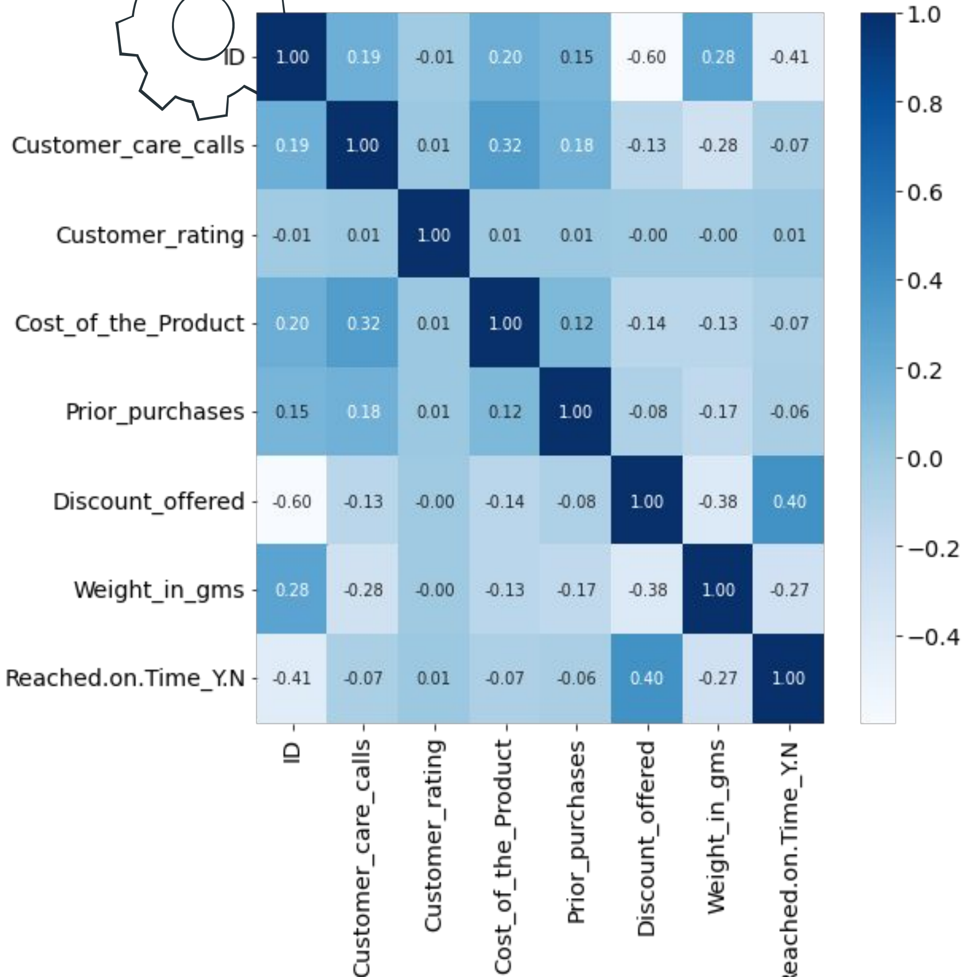



# Insight Discount Offer



# Data Understanding

## Feature



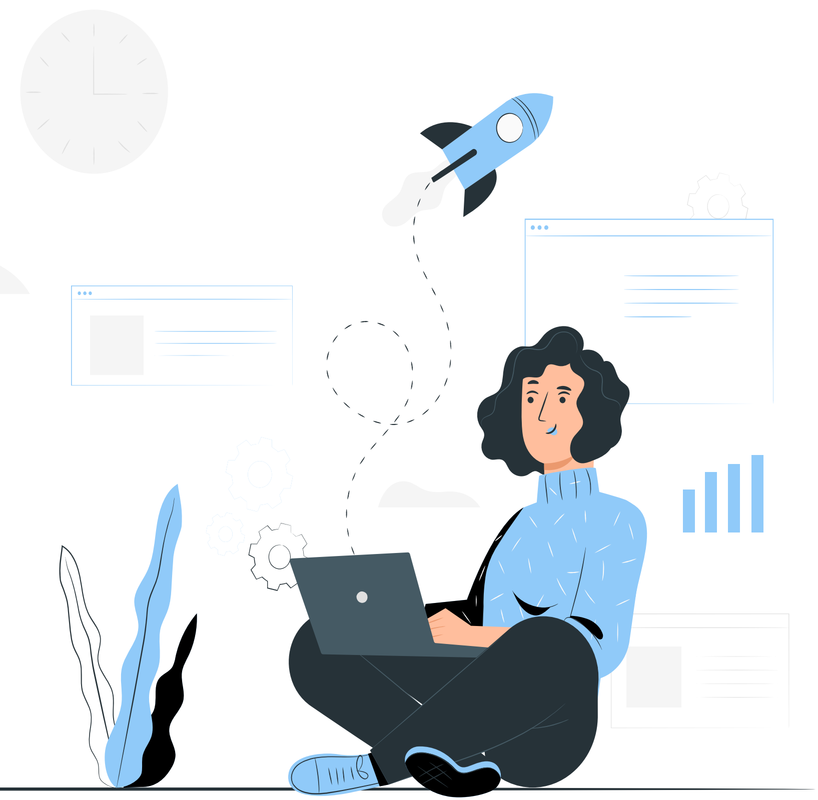

Dari correlation heatmap di samping dapat dilihat bahwa:

- Target kita Reached.on.Time_Y.N memiliki korelasi positif lemah dengan customer_rating, cost_of_the_product, customer_care_calls dan prior_purchases
- Ia juga memiliki korelasi positif cukup kuat dengan Discount_offered
- Ia juga memiliki korelasi negatif cukup kuat dengan weight_in_gms

Conclusion : Tidak terdapat fitur redundant karena tidak ada feature yang memiliki korelasi yang kuat diatas 0.7

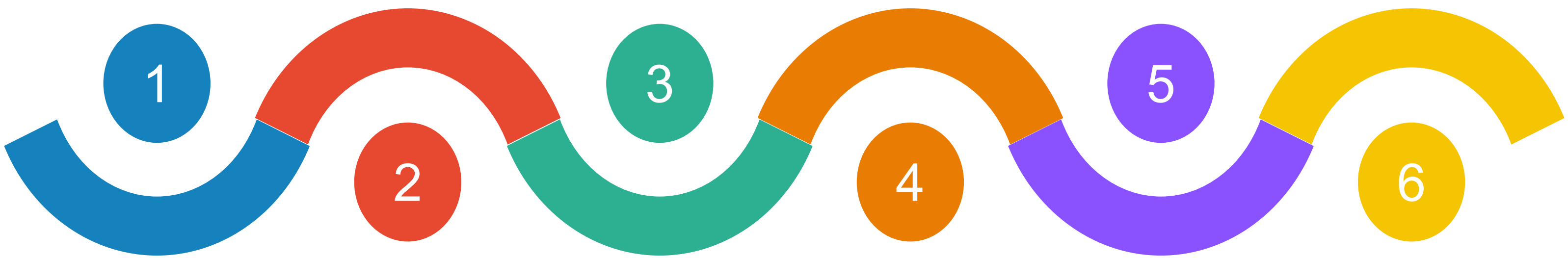
# 03

# Pre-Processing & Modelling



# 03 Pre-Processing & Modelling

## Pre-Processing




### Missing & Duplicated

**No missing and duplicated** values in dataset

### Outliers

Replace outliers in dataset using **IQR limit**

### Feature Transformation

**Standardization** for numerical feature

### Feature Encoding

**One Hot** and **Label Encoding** for categorical feature

### Class Imbalance

**No class imbalance** in target feature

### Feature Selection

Drop feature: **'ID', 'Customer_care_calls',** and **'Customer_rating'**

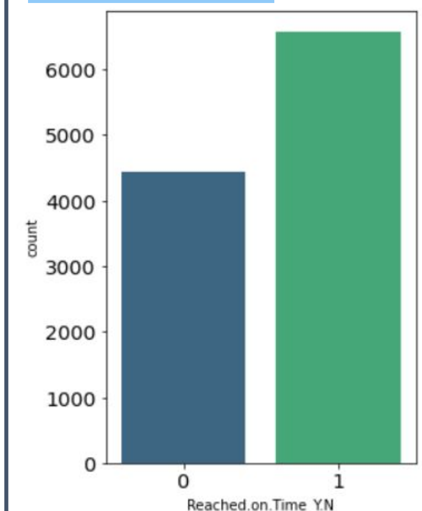
# 03 Pre-Processing & Modelling

## Pre-Processing

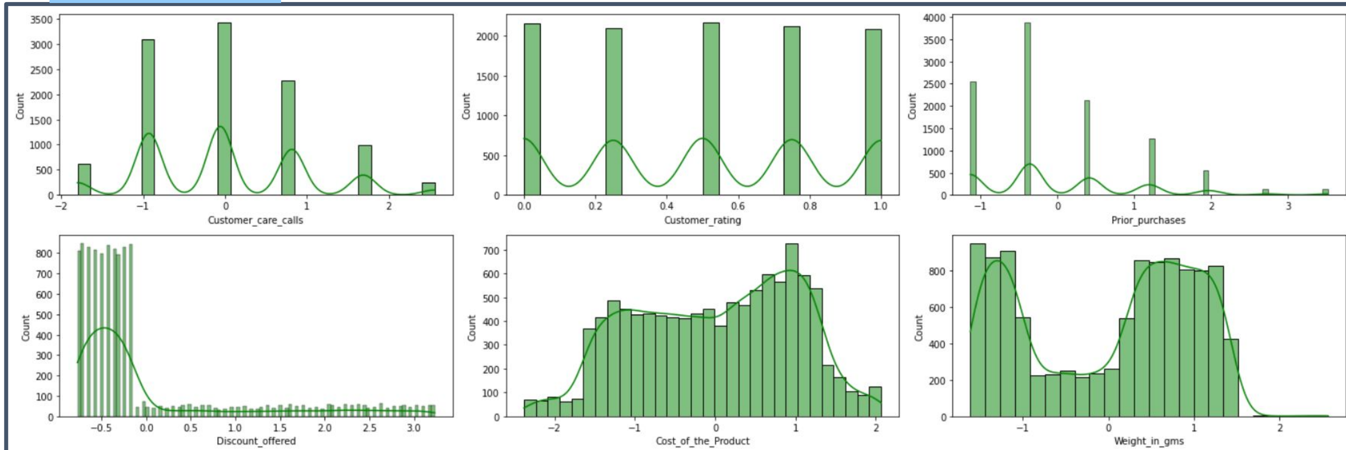
**Outliers .**

replaced Discount and Purcase outliers with IQR Limit

**Class imbalance .**




**Transformation .**




Standardization

Ratio of target feature

# 03 Pre-Processing & Modelling

## Pre-Processing

**Feature Transformation**

Standarisasi :

- Cost_of_the_Product
- Prior_purchases
- Discount_offered
- Weight_in_gms

**Encoding .**

`Product_importance` & `Gender` : Label Encoding

`Mode_of_Shipment` & `Warehouse_block` : One Hot Encoding

**Dataset for modelling (After Rename): .**

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10999 entries, 0 to 10998
Data columns (total 15 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Cost                    10999 non-null  float64
 1   Purchase                10999 non-null  float64
 2   Importance              10999 non-null  int64
 3   Gender                  10999 non-null  int64
 4   Discount                10999 non-null  float64
 5   Weight                  10999 non-null  float64
 6   Late                    10999 non-null  int64
 7   Mode_of_Shipment_Flight 10999 non-null  uint8
 8   Mode_of_Shipment_Road   10999 non-null  uint8
 9   Mode_of_Shipment_Ship   10999 non-null  uint8
 10  Warehouse_block_A       10999 non-null  uint8
 11  Warehouse_block_B       10999 non-null  uint8
 12  Warehouse_block_C       10999 non-null  uint8
 13  Warehouse_block_D       10999 non-null  uint8
 14  Warehouse_block_F       10999 non-null  uint8
dtypes: float64(4), int64(3), uint8(8)
memory usage: 687.6 KB
```

# 03 Pre-Processing & Modelling

## Modelling Result

**Best Modelling Result Before Feature Selection**

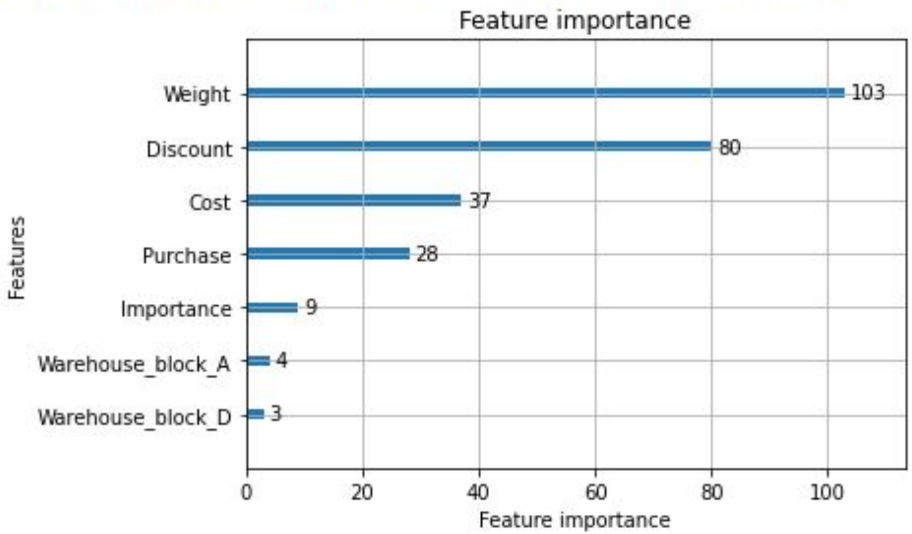
| | Decision Tree | Logistic Regression | LightGBM | KNN | Random Forest | XGBoost |
|---|---|---|---|---|---|---|
| Accuracy | 0.65 | 0.63 | 0.68 | 0.65 | 0.66 | 0.69 |
| Precision | 0.74 | 0.68 | 0.79 | 0.72 | 0.74 | 0.89 |
| Recall | 0.62 | 0.71 | 0.63 | 0.68 | 0.65 | 0.55 |
| F1-Score | 0.68 | 0.70 | 0.70 | 0.70 | 0.69 | 0.68 |
| ROC-AUC | 0.65 | 0.62 | 0.75 | 0.65 | 0.66 | 0.72 |

**Primary : ROC-AUC**
**Secondary : F1-Score**

# 03 Pre-Processing & Modelling
## Interpretation

**Feature importance LightGBM .**

<matplotlib.axes._subplots.AxesSubplot at 0x7f887a796910>




**Top 4 Feature**

| Feature | Correlation |
|---|---|
| Discount_offered | **0.40** |
| Weigth_in_gms | **-0.27** |
| Cost_of_Product | **-0.07** |
| Prior_purchases | **-0.06** |

**Top 4 feature show direct relationship to the target ‘Reached.on.Time_Y.N or ‘Late”**

# 03 Pre-Processing & Modelling

## Modelling Result

**Best Modelling Result After Feature Selection**

| | Decision Tree | Logistic Regression | LightGBM | KNN | Random Forest | XGBoost |
|---|---|---|---|---|---|---|
| Accuracy | 0.68 | 0.64 | 0.69 | 0.66 | 0.69 | 0.69 |
| Precision | 0.87 | 0.69 | 0.89 | 0.80 | 0.94 | 0.89 |
| Recall | 0.54 | 0.72 | 0.55 | 0.58 | 0.51 | 0.54 |
| F1-Score | 0.67 | 0.70 | 0.67 | 0.67 | 0.66 | 0.67 |
| ROC-AUC | 0.71 | 0.62 | 0.75 | 0.68 | 0.73 | 0.72 |

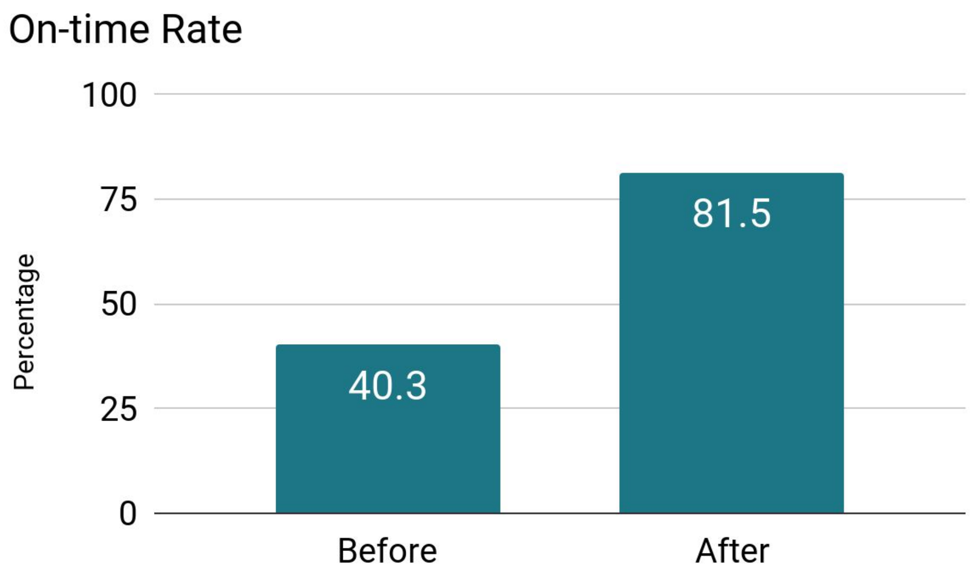
**Primary : ROC-AUC**
**Secondary : F1-Score**

# 04

# Recommendations

# 04 Recommendation

## On-Time Rate




On time Rate Mengalami peningkatan sebanyak 102.1% dari yang sebelumnya 40.3% menjadi 81.5% berdasarkan predictive modelling

# 04 Recommendation

## Hypothetical Loss Saved

**$196.82**

Avg revenue / customer

**$1,291,729.66**

Potential revenue loss

Jika customer berhenti belanja, maka kemungkinan kerugian pendapatan yang dialami perusahaan sebanyak **$1.3 Million**

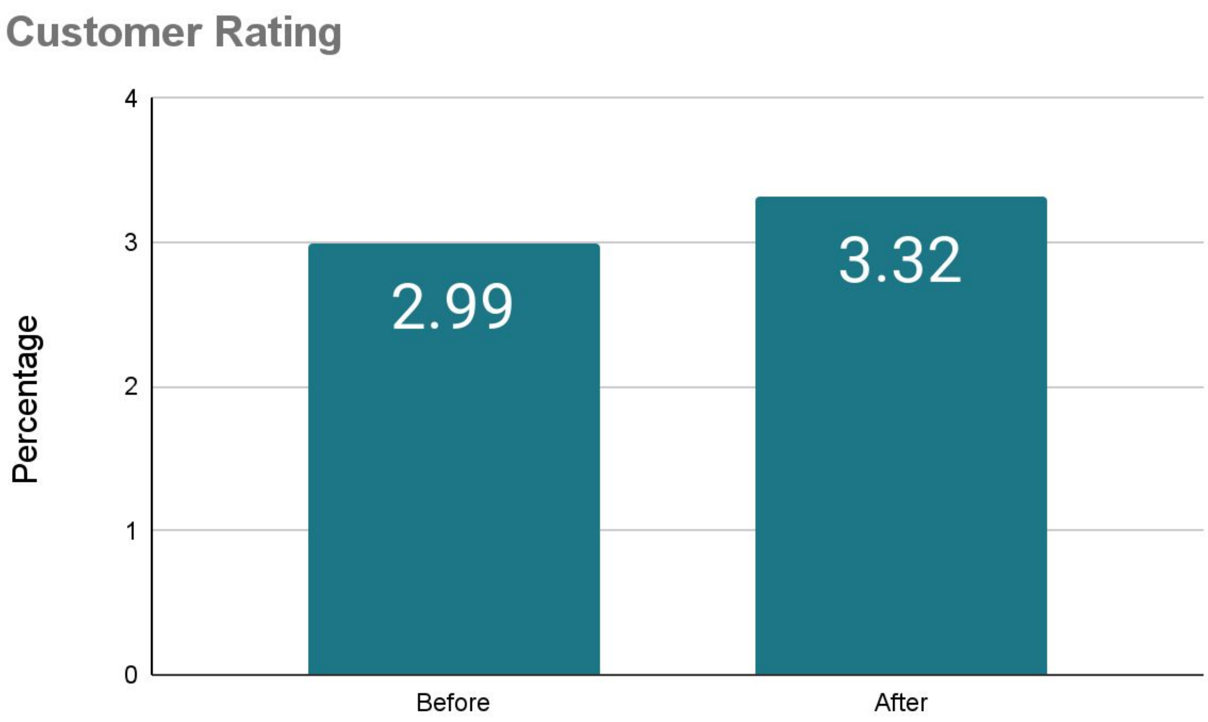
**$891,200.96** **69%**

Potential revenue loss saved

Tapi dengan menggunakan predictive modelling perusahaan dapat menghemat sampai dengan **$891 thousand**

# 04 Recommendation

## Customer Satisfaction




Model kita memberikan peningkatan sebanyak 11% dalam customer rating

Ini dibuktikan dengan rata-rata rating sebelumnya 2.99% menjadi 3.32%

Ini dapat terjadi dengan memberikan bintang 1 untuk setiap keterlambatan kecuali untuk pelanggan yang telah memberikan bintang 5, karena bintang 5 adalah nilai maksimal yang dapat diberikan

# 04 Recommendation

## Business Recommendations

### SHORT TERM

**Notify and compensate customers if packages are late**

Customers will have higher satisfaction if they are given updates and compensation on their packages

**Do an internal audit**

Many entities involved in the shipping process are underperforming and it is worth investigating the reason

### LONG TERM

**Add more relevant dimensions**

Measurements such as package departure time, and distance will provide better insight and analysis

**Develop strategies based on influential factors**

Strategies based around package weight and discount may help optimize the shipping process w3.solutions()

# THANK YOU!

# APPENDIX

# Data Exploration

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10999 entries, 0 to 10998
Data columns (total 12 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   ID                   10999 non-null  int64 
 1   Warehouse_block      10999 non-null  object
 2   Mode_of_Shipment     10999 non-null  object
 3   Customer_care_calls  10999 non-null  int64 
 4   Customer_rating      10999 non-null  int64 
 5   Cost_of_the_Product  10999 non-null  int64 
 6   Prior_purchases      10999 non-null  int64 
 7   Product_importance   10999 non-null  object
 8   Gender               10999 non-null  object
 9   Discount_offered     10999 non-null  int64 
 10  Weight_in_gms        10999 non-null  int64 
 11  Reached.on.Time_Y.N  10999 non-null  int64 
dtypes: int64(8), object(4)
memory usage: 1.0+ MB
```

Data Understanding and Describe

**10999** Total rows

**0** Total null value

```
# pengelompokan kolom berdasarkan jenisnya
nums = ['Customer_care_calls', 'Customer_rating', 'Prior_purchases', 'Discount_offered', 'Cost_of_the_Product', 'Weight_in_gms', 'Reached.on.Time_Y.N
cats = ['Mode_of_Shipment', 'Product_importance', 'Gender','Warehouse_block']
```

# Data Statistics Description

| | Customer_care_calls | Customer_rating | Prior_purchases | Discount_offered | Cost_of_the_Product | Weight_in_gms | Reached.on.Time_Y.N |
|---|---|---|---|---|---|---|---|
| count | 10999.000000 | 10999.000000 | 10999.000000 | 10999.000000 | 10999.000000 | 10999.000000 | 10999.000000 |
| mean | 4.054459 | 2.990545 | 3.567597 | 13.373216 | 210.196836 | 3634.016729 | 0.596691 |
| std | 1.141490 | 1.413603 | 1.522860 | 16.205527 | 48.063272 | 1635.377251 | 0.490584 |
| min | 2.000000 | 1.000000 | 2.000000 | 1.000000 | 96.000000 | 1001.000000 | 0.000000 |
| 25% | 3.000000 | 2.000000 | 3.000000 | 4.000000 | 169.000000 | 1839.500000 | 0.000000 |
| 50% | 4.000000 | 3.000000 | 3.000000 | 7.000000 | 214.000000 | 4149.000000 | 1.000000 |
| 75% | 5.000000 | 4.000000 | 4.000000 | 10.000000 | 251.000000 | 5050.000000 | 1.000000 |
| max | 7.000000 | 5.000000 | 10.000000 | 65.000000 | 310.000000 | 7846.000000 | 1.000000 |

## Beberapa Pengamatan

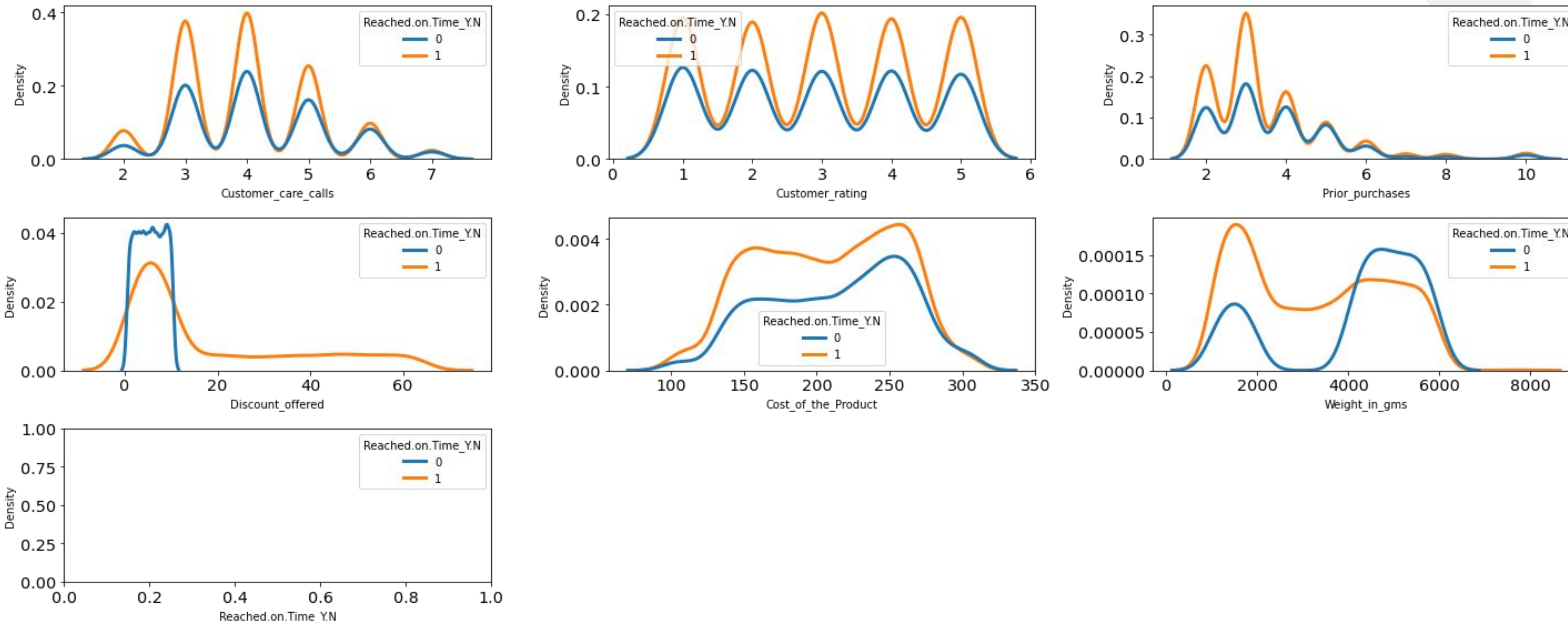
- Kolom Customer_care_calls, customer_rating, dan Cost_of_the_Product tampak sudah cukup simetrik distribusinya (mean dan median tak berbeda jauh)
- Kolom Discount_offered dan Prior_purchases tampaknya skew ke kanan (long-right tail)
- Kolom Reached.on.Time_Y.Nbernilai boolean/binary

| | Mode_of_Shipment | Product_importance | Gender | Warehouse_block |
|---|---|---|---|---|
| count | 10999 | 10999 | 10999 | 10999 |
| unique | 3 | 3 | 2 | 5 |
| top | Ship | low | F | F |
| freq | 7462 | 5297 | 5545 | 3666 |

- Untuk kategori gender perempuan lebih dominan,
- untuk kategori product importance di dominasi oleh kategori low
- untuk kategori mode pengiriman di dominasi oleh pengiriman menggunakan kapal (ship)
- untuk warehouse_block didominasi oleh block F
- Semua unique value tiap kategori masih dalam kategori normal sekitar 2-5 unique values

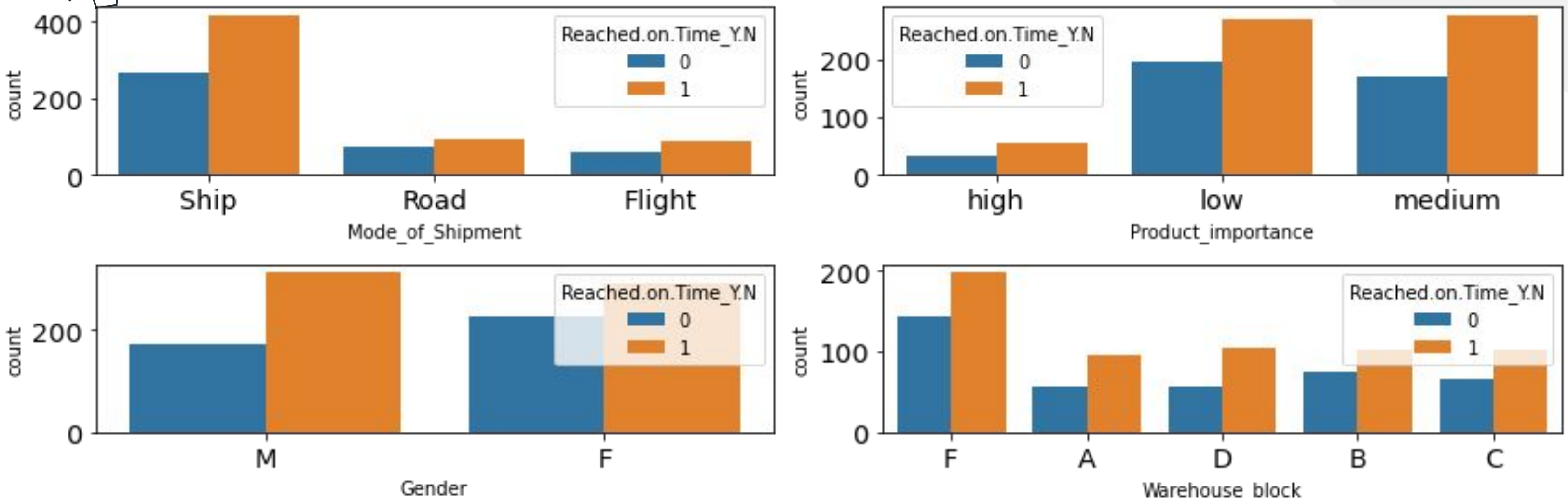
# Data Understanding

## Numerical Feature



# Data Understanding

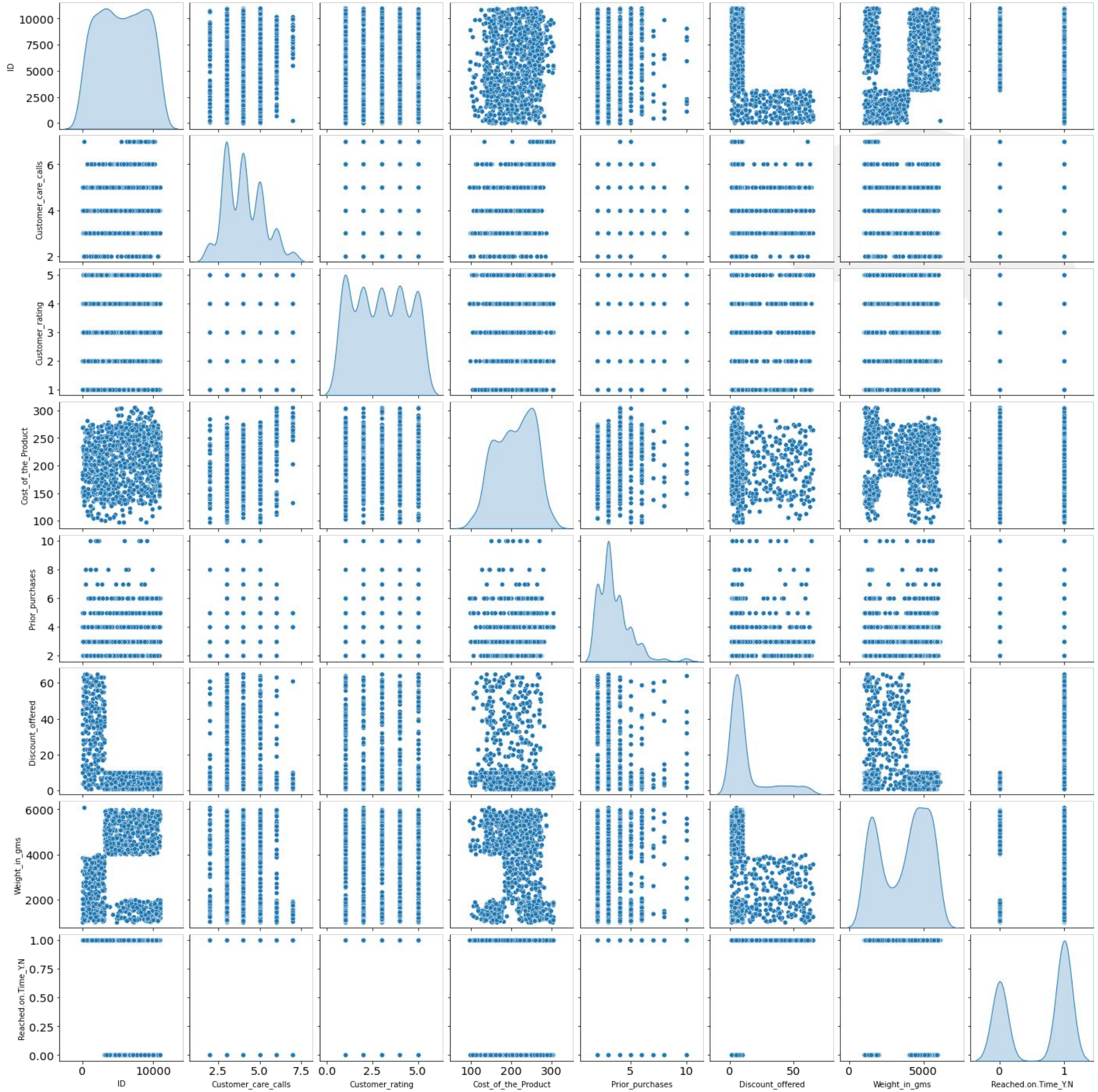
## Catregorical Feature




Pengamatan

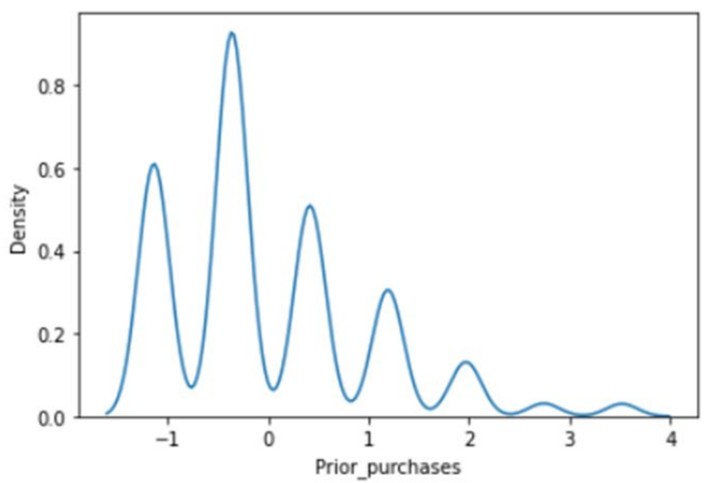
- shipment dengan ship cenderung akan mengalami telat pengiriman
- untuk produk_importance dengan kategori low dan medium cenderung akan mengalami telat pengiriman
- untuk warehouse_block dengan kategori F cenderung mengalami telat pengirimanPengamatan
- shipment dengan ship cenderung akan mengalami telat pengiriman
- untuk produk_importance dengan kategori low dan medium cenderung akan mengalami telat pengiriman
- untuk warehouse_block dengan kategori F cenderung mengalami telat pengiriman

# Data Understanding



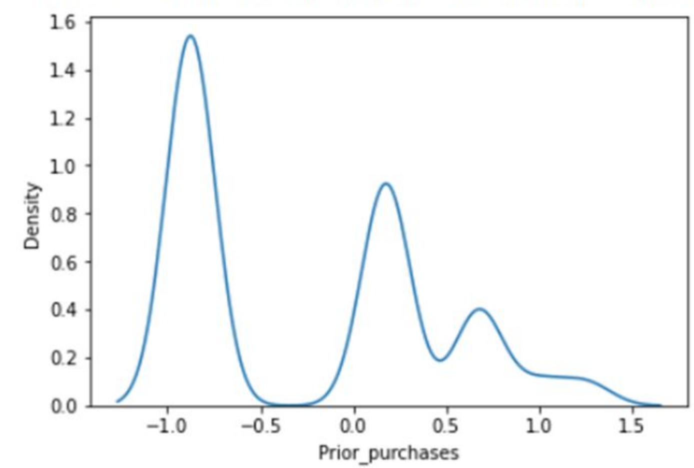
# Log Transformation

```
sns.kdeplot(df2['Prior_purchases']);
```



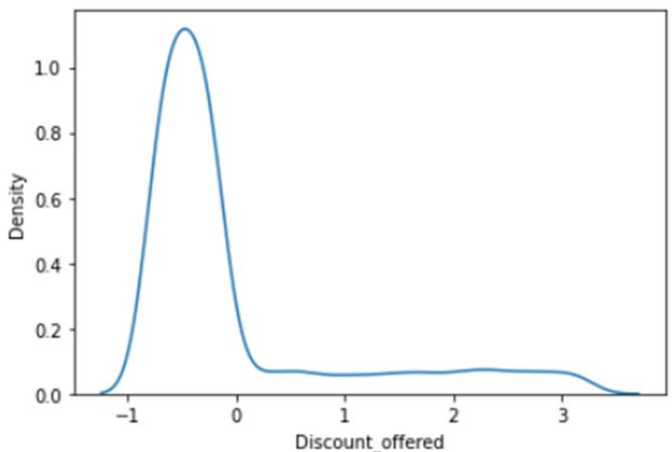

```
sns.kdeplot(np.log(df2['Prior_purchases']));
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/arra
  result = getattr(ufunc, method)(*inputs, **kwargs)
```



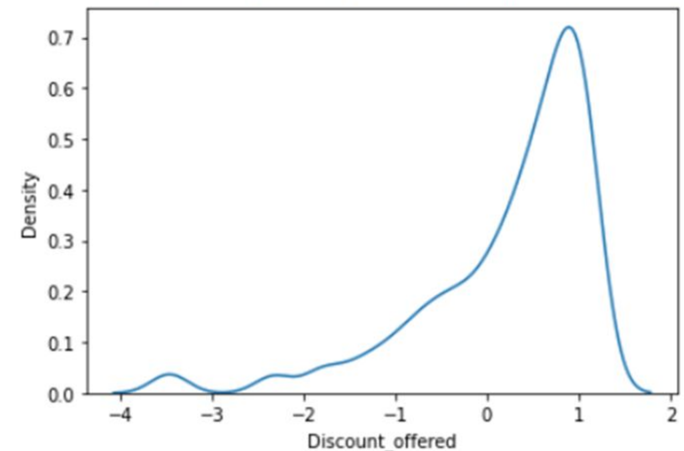
# Log Transformation

```
sns.kdeplot(df2['Discount_offered']);
```



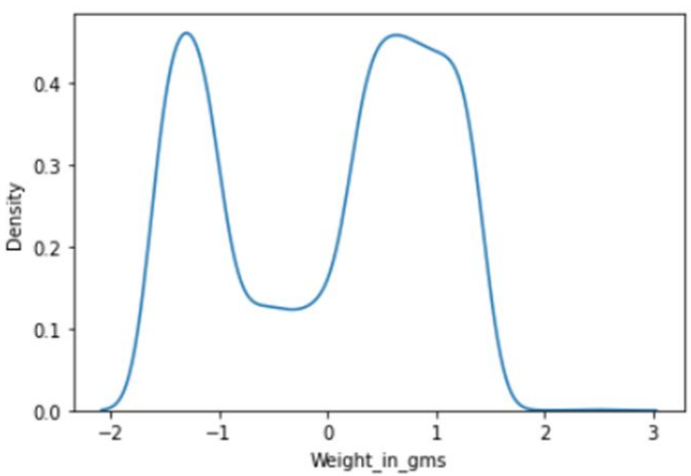

```
sns.kdeplot(np.log(df2['Discount_offered']));
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/arr
  result = getattr(ufunc, method)(*inputs, **kwargs)
```



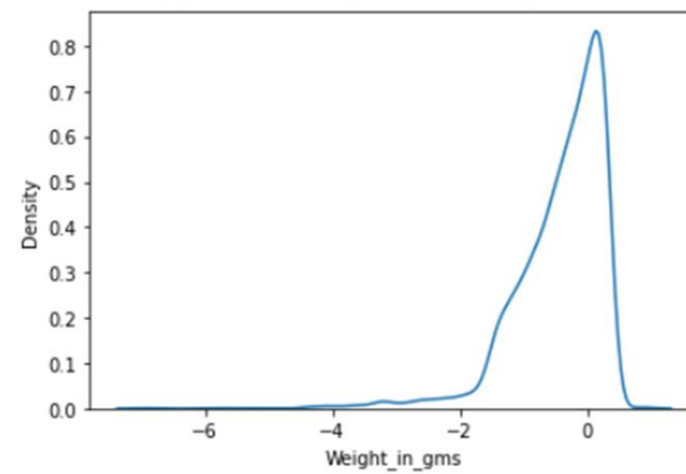
# Log Transformation

```
sns.kdeplot(df2['Weight_in_gms']);
```



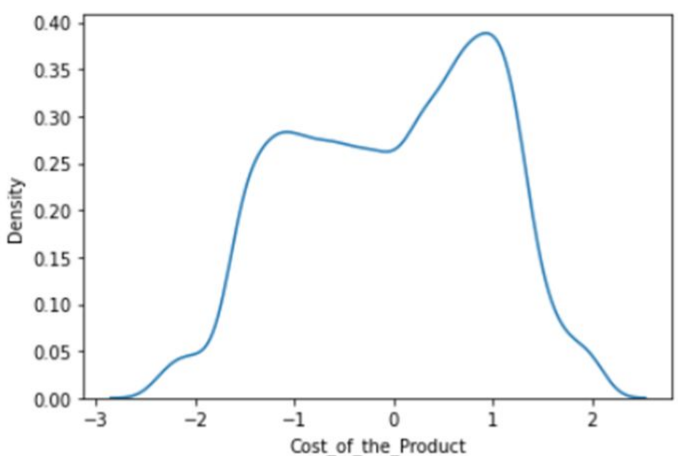

```
sns.kdeplot(np.log(df2['Weight_in_gms']));
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/arra
  result = getattr(ufunc, method)(*inputs, **kwargs)
```



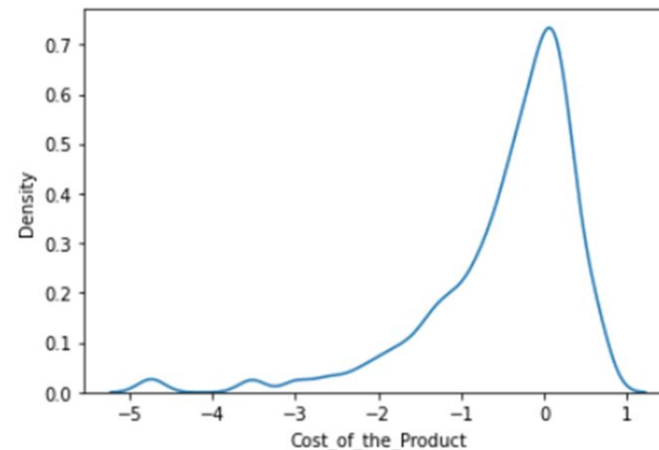
# Log Transformation

```
sns.kdeplot(df2['Cost_of_the_Product']);
```



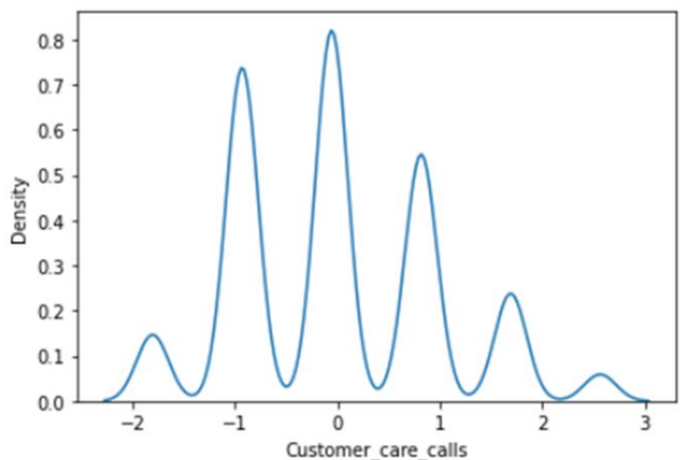

```
sns.kdeplot(np.log(df2['Cost_of_the_Product']));
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/arra
  result = getattr(ufunc, method)(*inputs, **kwargs)
```



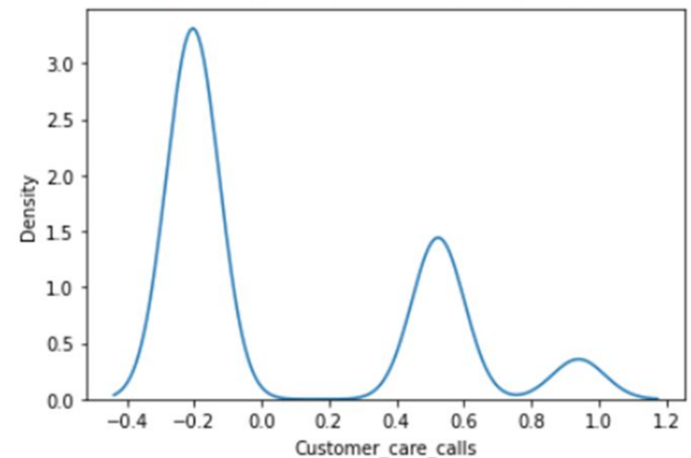
# Log Transformation

```
sns.kdeplot(df2['Customer_care_calls']);
```




```
sns.kdeplot(np.log(df2['Customer_care_calls']));
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/arra
  result = getattr(ufunc, method)(*inputs, **kwargs)
```



# EDA Conclusion

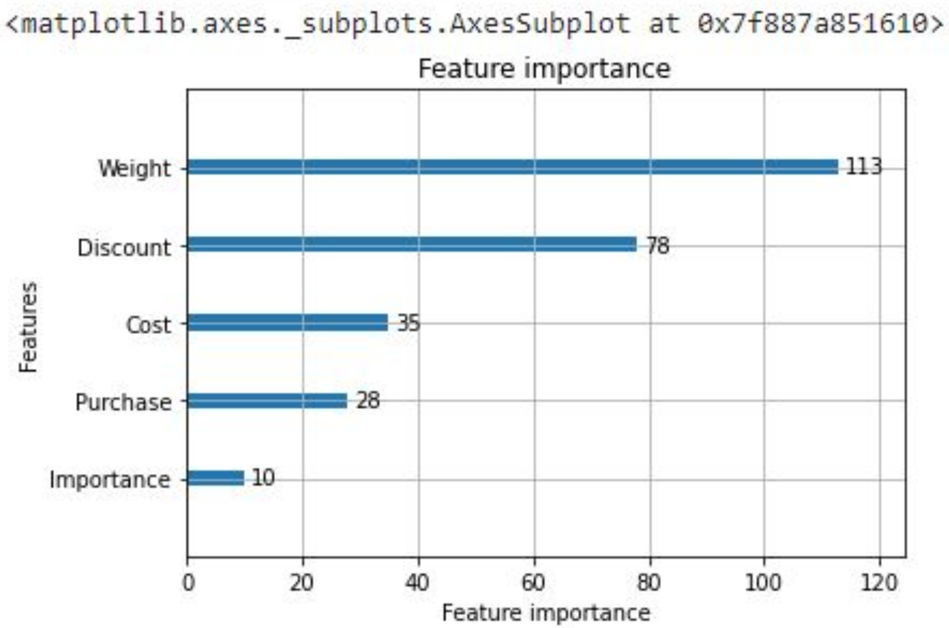
Beberapa hal yang kita temukan dari EDA dataset ini adalah:

- Data terlihat valid dan tidak ada kecacatan yang major/signifikan
- Ada beberapa distribusi yang sedikit skewed, hal ini harus diingat apabila kita ingin melakukan sesuatu atau menggunakan model yang memerlukan asumsi distribusi normal
- Beberapa feature memiliki korelasi yang jelas dengan target, mereka akan dipakai
- Beberapa feature terlihat sama sekali tidak berkorelasi, mereka sebaiknya diabaikan
- Dari fitur kategorikal, “mode_of_shipment” ,” warehouse_block”, dan “product_importance” sepertinya berguna untuk menjadi prediktor model

# Pre-Processing & Modelling

## Interpretation

**Feature importance LightGBM after feature selection**




**Top 4 Feature**

| Feature | Correlation |
|---|---|
| Discount_offered | **0.40** |
| Weigth_in_gms | **-0.27** |
| Cost_of_Product | **-0.07** |
| Prior_purchases | **-0.06** |

**Top 4 feature show direct relationship to the target 'Reached.on.Time_Y.N'**

# Confusion Matrix

**Modelling Result By LightGBM**

**Actual Value**

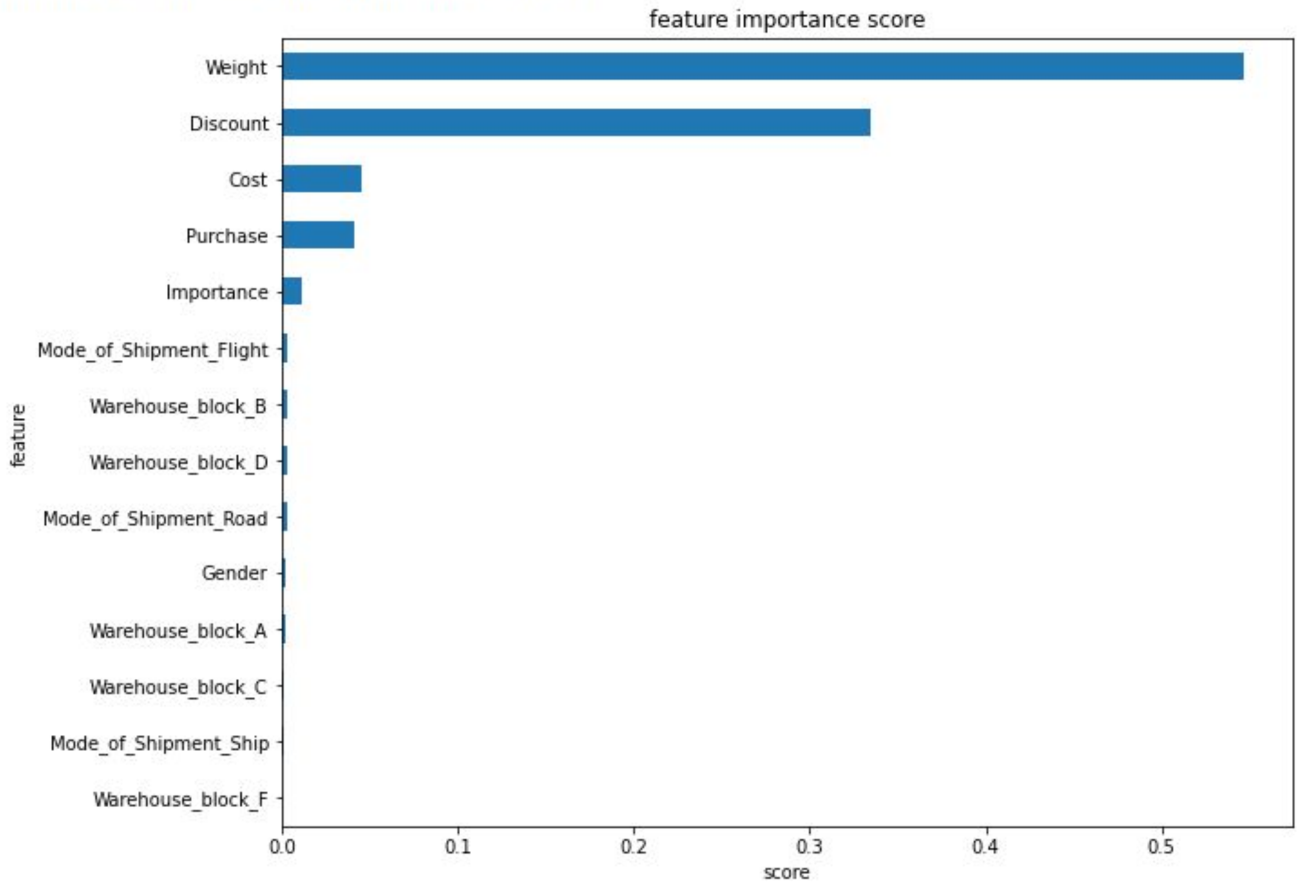
**Predicted value**

| | Positive | Negative |
|---|---|---|
| Positive | **812 (TP)** | **68 (FP)** |
| Negative | **615 (FN)** | **690 (TN)** |

# Feature Importance

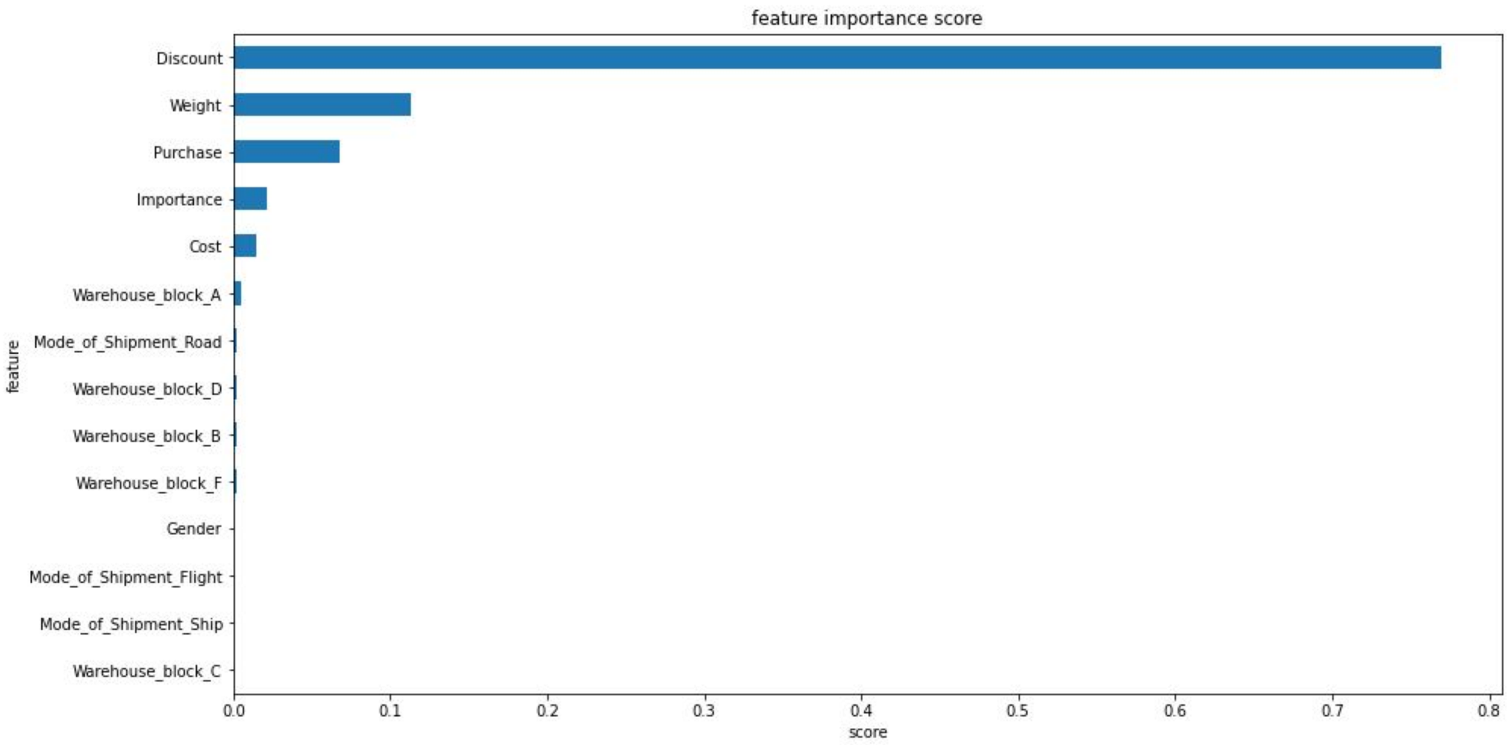
## Random Forest

```
Text(0.5, 1.0, 'feature importance score')
```



# Feature Importance

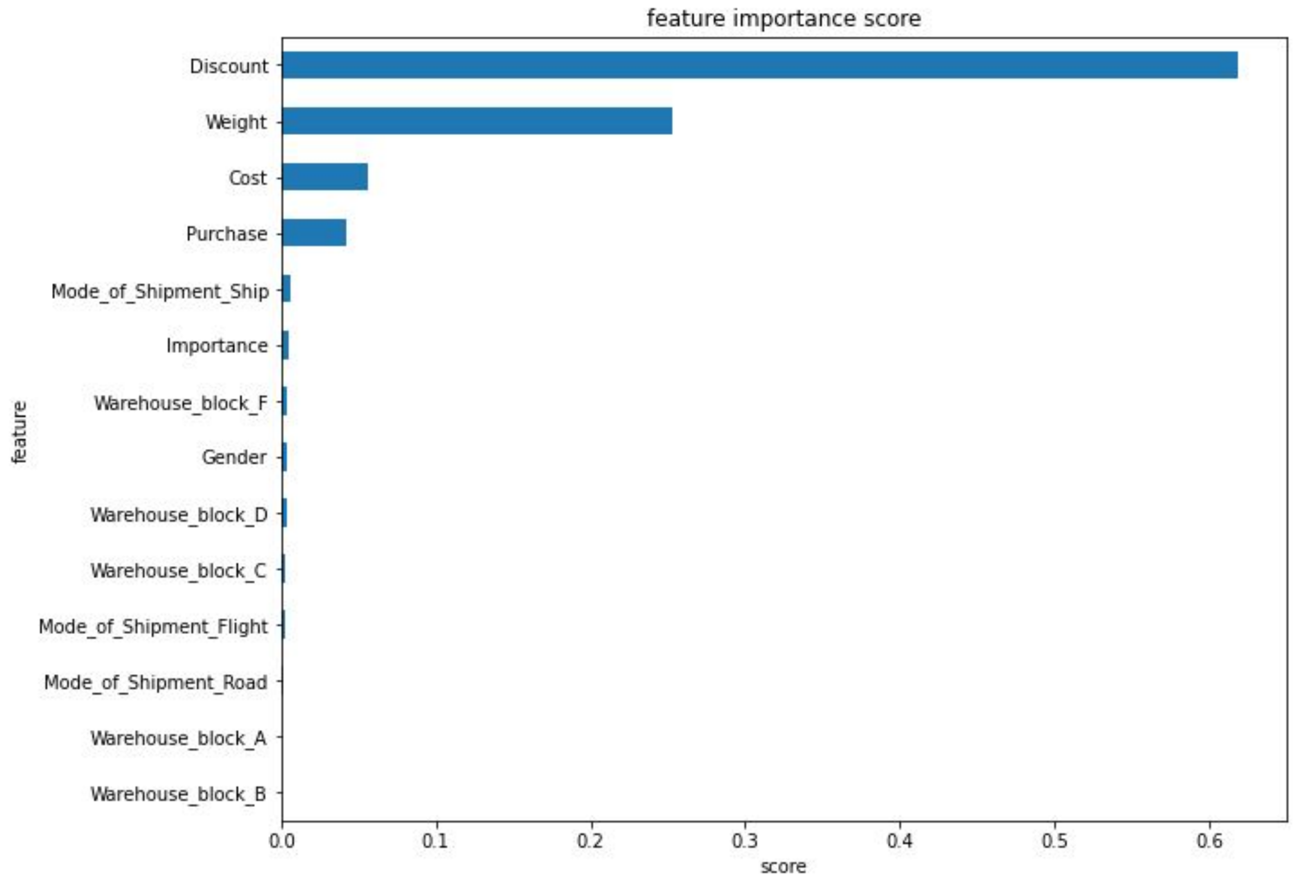
## XGBoost



# Feature Importance

## Decision Tree

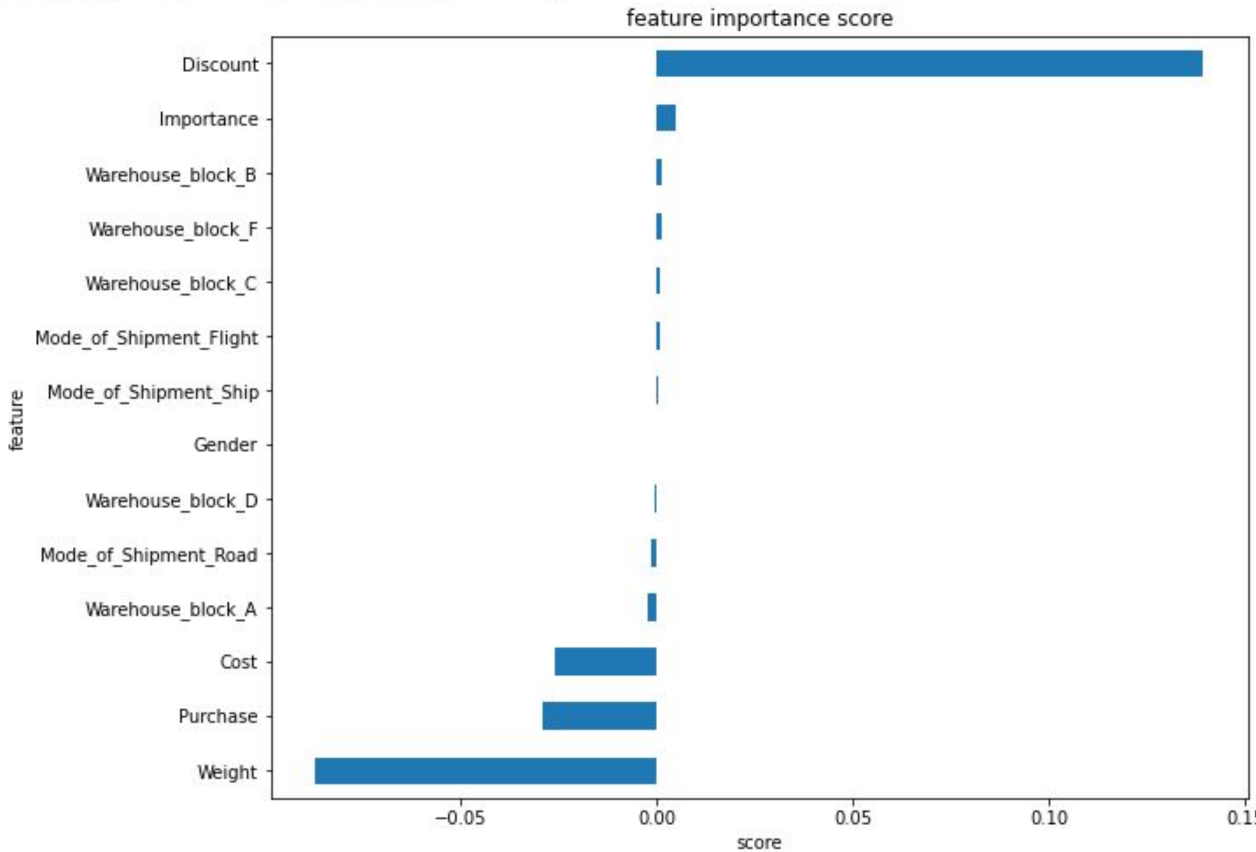
```
Text(0.5, 1.0, 'feature importance score')
```



# Feature Importance

## Logistic Regression

```
Text(0.5, 1.0, 'feature importance score')
```



# Modelling Result

**Modelling without feature selection and without hyperparameter**

| | Decision Tree | Logistic Regression | Lightgbm | KNN | Random Forest | XGBoost |
|---|---|---|---|---|---|---|
| Accuracy | 0.65 | 0.63 | 0.68 | 0.65 | 0.66 | 0.69 |
| Precision | 0.70 | 0.68 | 0.79 | 0.72 | 0.74 | 0.89 |
| Recall | 0.71 | 0.71 | 0.63 | 0.68 | 0.65 | 0.55 |
| F1-Score | 0.71 | 0.70 | 0.70 | 0.70 | 0.69 | 0.68 |
| ROC-AUC | 0.64 | 0.62 | 0.75 | 0.65 | 0.66 | 0.72 |

**Primary : ROC-AUC**
**Secondary : F1-Score**

# Modelling Result

Modelling without feature selection and with hyperparameter

| | Decision Tree | Logistic Regression | Lightgbm | KNN | Random Forest | XGBoost |
|---|---|---|---|---|---|---|
| Accuracy | 0.65 | 0.59 | 0.69 | 0.65 | 0.62 | 0.65 |
| Precision | 0.74 | 0.59 | 0.9 | 0.72 | 0.63 | 0.70 |
| Recall | 0.62 | 1.00 | 0.53 | 0.68 | 0.85 | 0.72 |
| F1-Score | 0.68 | 0.74 | 0.67 | 0.70 | 0.73 | 0.71 |
| ROC-AUC | 0.65 | 0.50 | 0.75 | 0.65 | 0.57 | 0.63 |

**Primary : ROC-AUC**
**Secondary : F1-Score**

# Modelling Result

Modelling with feature selection and without hyperparameter

| | Decision Tree | Logistic Regression | Lightgbm | KNN | Random Forest | XGBoost |
|---|---|---|---|---|---|---|
| Accuracy | 0.65 | 0.64 | 0.67 | 0.65 | 0.67 | 0.69 |
| Precision | 0.70 | 0.69 | 0.78 | 0.72 | 0.74 | 0.89 |
| Recall | 0.71 | 0.72 | 0.62 | 0.68 | 0.68 | 0.54 |
| F1-Score | 0.71 | 0.70 | 0.69 | 0.70 | 0.71 | 0.67 |
| ROC-AUC | 0.63 | 0.62 | 0.74 | 0.65 | 0.67 | 0.72 |

**Primary : ROC-AUC**
**Secondary : F1-Score**

# Modelling Result

Modelling with feature selection and with hyperparameter

| | Decision Tree | Logistic Regression | Lightgbm | KNN | Random Forest | XGBoost |
|---|---|---|---|---|---|---|
| Accuracy | 0.68 | 0.59 | 0.69 | 0.65 | 0.69 | 0.66 |
| Precision | 0.87 | 0.59 | 0.89 | 0.72 | 0.94 | 0.75 |
| Recall | 0.54 | 1.00 | 0.55 | 0.68 | 0.51 | 0.64 |
| F1-Score | 0.67 | 0.74 | 0.67 | 0.70 | 0.66 | 0.69 |
| ROC-AUC | 0.71 | 0.50 | 0.75 | 0.65 | 0.73 | 0.67 |

**Primary : ROC-AUC**
**Secondary : F1-Score**