# Shipment Arrival Prediction

By : DataSquad

ELECTROUX

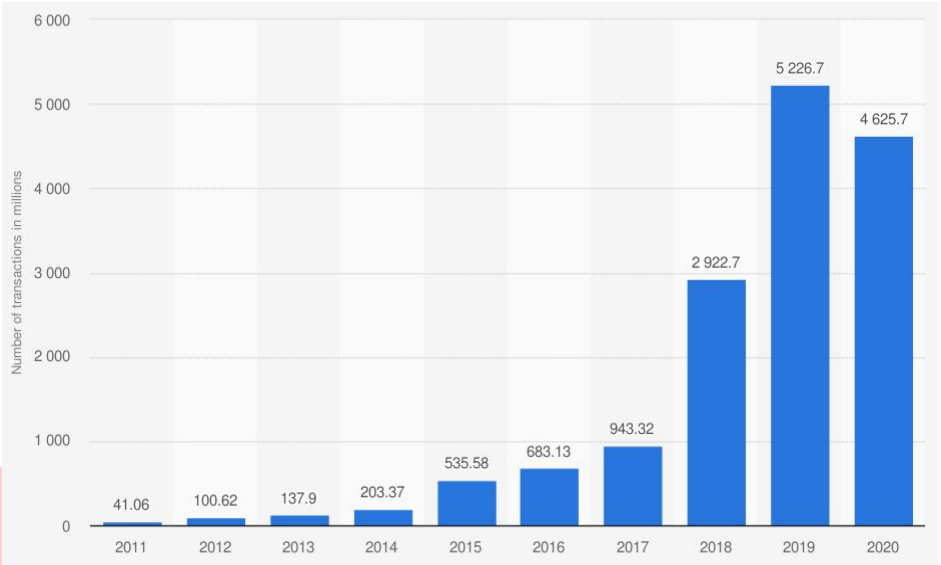
# ELECTROUX

An e-commerce based in Indonesia that focuses on providing various electronic products

Established in 2010,
**Electroux** is supported by **5 warehouses**
to support its operations that located around
**Jakarta area**

# Expansion: New Business Line

**Number of Online Transaction in Indonesia from 2011 to 2020 (in millions)**




Source: Statista, 2021

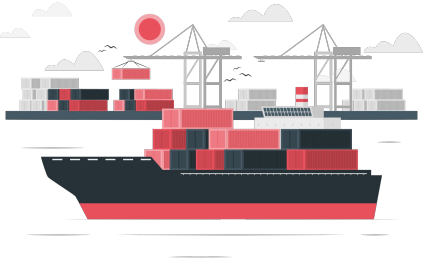
ELECTROUX

In 2018, the e-commerce decided to open a new business line which is the **inhouse logistics**

# Inhouse Logistic:
## Mode of Transportation

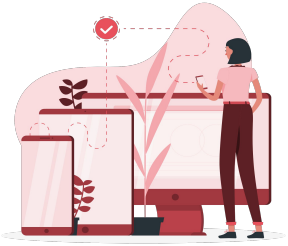
Land

Air

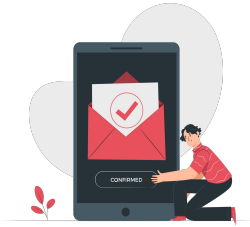


Sea

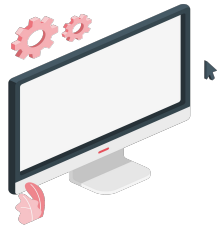
# Shipping Process



Customer finalize order




Ecommerce received order



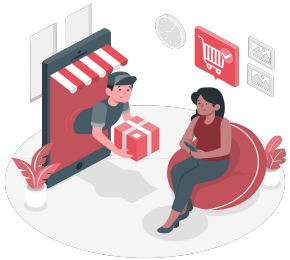
Ecommerce check product availability in warehouse

Order sent from the nearest warehouse that have available product

Order sent via air/land/sea
**Estimated time: 2-4 Days**

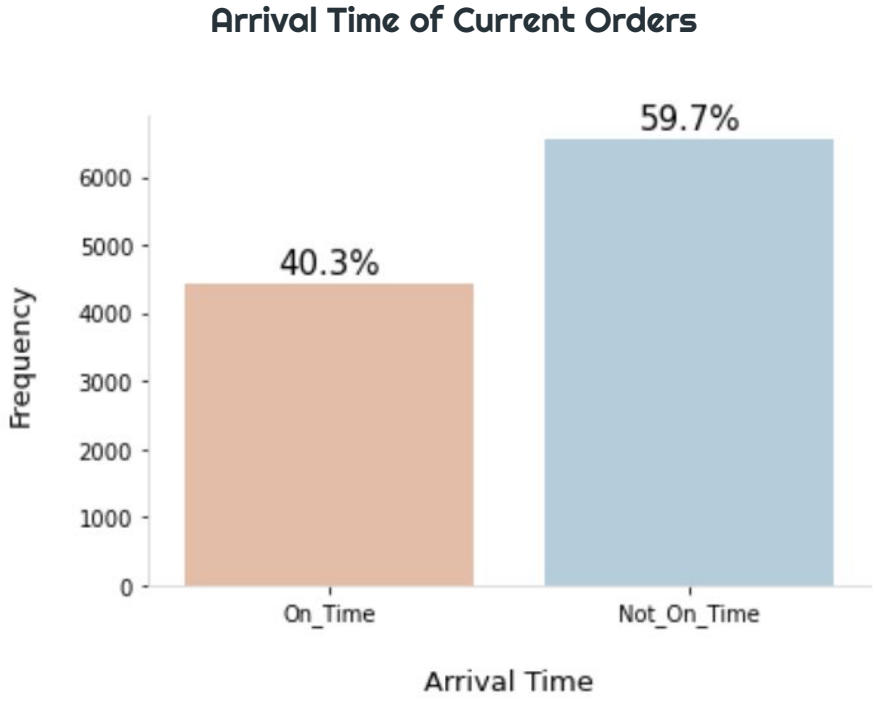
Order arrived in destination area

Order delivered to destination transit warehouse



Order are delivered to the customers

# Problem Identification




# Problem Identification

**35%** **of consumers**

see shipment as the biggest problem in e-commerce

**>90%** **of complaints**

related with late shipment or miscommunication about the shipment status (detik.com, 2019)

# Why Shipping Delays Negatively Impact E-commerce?

**69%** **of consumers**

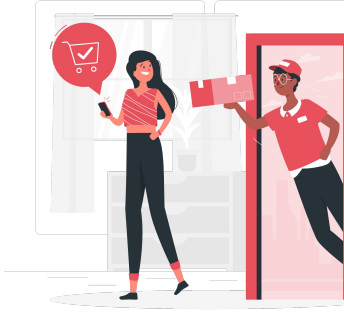
Less likely to shop in an e-commerce in the future if the package they purchased is not delivered within two days of the date promised

**17%** **of consumers**

Will stop shopping with an e-commerce after receiving a late delivery once

**55%** **of consumers**

Will stop shopping with an e-commerce after receiving a late delivery two or three late deliveries



Source: Jules, 2020

# Objective

Create a system to help predict & calculate late shipment automatically

# Business Metrics

Reduce **Late Shipment**

Reduce number of occurrence of late shipment

Reduce **Churned Customers**

Estimate the percentage of customers that most likely to churned

Increase **Potential Gross Profit**

Estimate potential gross profit earned by Electroux

# Dataset Overview

## Dataset

Historical Orders of Customers and Their Arrival Time Data
10,999 rows & 12 features

## Features

| | | | |
|---|---|---|---|
| Reach_on_time.Y.N | ID | Warehouse_block | Mode_of_Shipment |
| Customer_care_calls | Customer_rating | Cost_of_the_product | Prior_purchases |
| Product_importance | Gender | Discount_offered | Weight_in_gms |

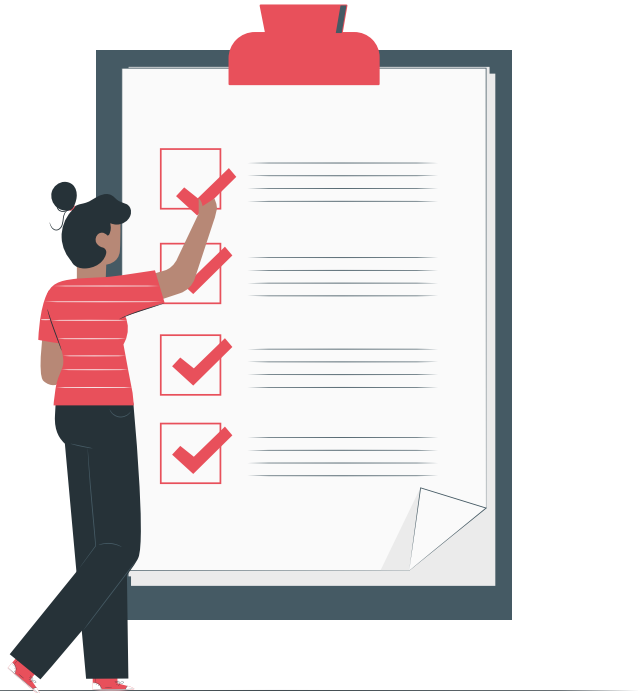
# Data Preprocessing

## 1. Data Cleansing

No missing data
No duplicates
Drop Feature 'ID'

## 2. Standardization

Feature 'Customer_care_calls'
Feature 'Customer_rating'
Feature 'Cost_of_the_product'



## 3. Normalization

Feature 'Prior_purchases'
Feature 'Discount_offered'
Feature 'Weight_in_gms'

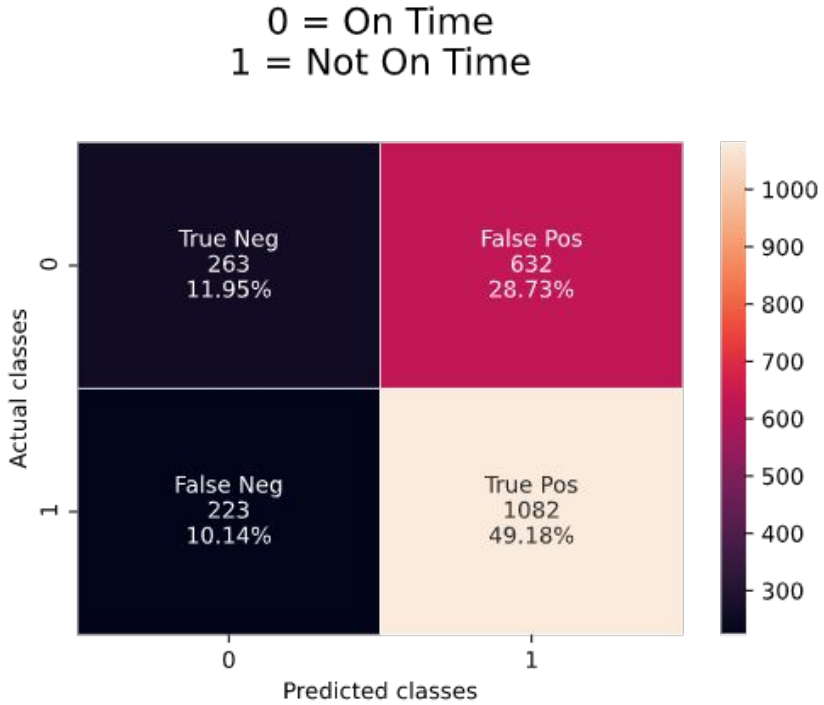
## 4. Feature Encoding

**One-Hot Encoding**
Feature 'Warehouse_block'
Feature 'Mode_of_Shipment'
Feature 'Gender'

**Label Encoding**
Feature 'Product_importance'

# Model Evaluation Results

## Reduce Late Shipment



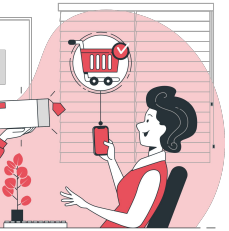

*Recall* : trying to minimize false negative
true positive / (true positive+false negative)

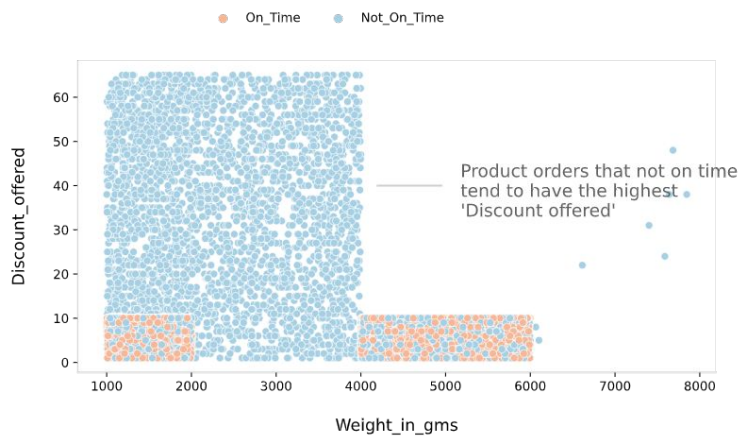
- We are going to focus on ***recall*** (*minimize actual not on time, but predicted on time*) to evaluate the model performance
- The best prediction performance score using ***recall*** metric is **83%**
- The feature that influences the most is the '**discount offered**'

# Recommendation

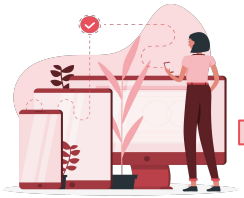
We recommend that:

**Electroux** to provide a **new shipping option (Express - 1 Day)**
for shipments that predicted to arrive late



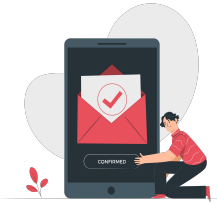



# Shipping Process
## (after Machine Learning)

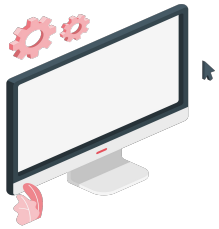


Customer finalize order


**Shipment Option**:
- Reguler
- Express



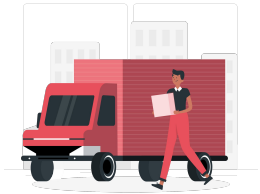
Ecommerce received order



Ecommerce check product availability in warehouse

Order sent from the nearest warehouse that have available product

Order sent via air/land/sea



Order arrived in destination area

Order delivered to destination transit warehouse

Order are delivered to the customers

# Comparison Before & After Model Modification

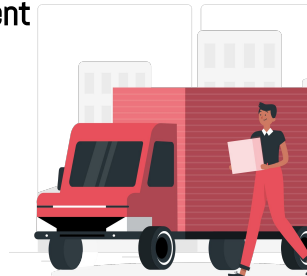
## Reduce Churned Customers

**Before Modification :**

60% Late Shipment x 17% Potential Churned Customer = 10,2% Potential Churned Customers from Late Shipment

**After Modification :**

60% Late Shipment x 17% Potential Failure To Predict Late Shipment x 17% Potential Churned Customer = 1,734% Potential Churned Customers from Late Shipment



# Potential profit


The prediction of potential losses be overcome by **Electroux** is

**~3 billion rupiah**

# Conclusions

**Number of occurence of late shipment** the by performance is **83%**

**Percentage of churned customers** by performance is reduce **8,466%**

We recommend that **Electroux** to provide a **new shipping option (Express - 1 Day)** for shipments that predicted to arrive late

The prediction of potential profit earned by **Electroux** is

**~3 billion rupiah**

# Thank You~

## Shipment Arrival Prediction

By : DataSquad

# Appendix

# 1. Data Extraction

## Data Exploration

```
data.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
```

```
m = data.shape[0]
n = data.shape[1]

print("Number of rows: " + str(m))
print("Number of columns: " + str(n))
```

```
Number of rows: 10999
Number of columns: 12
```

## Data Cleansing

```
# check missing values
data.isnull().sum()
```

```
ID                     0
Warehouse_block        0
Mode_of_Shipment       0
Customer_care_calls    0
Customer_rating        0
Cost_of_the_Product    0
Prior_purchases        0
Product_importance     0
Gender                 0
Discount_offered       0
Weight_in_gms          0
Reached.on.Time_Y.N    0
dtype: int64
```

There are no ***missing values*** in all columns and ***no duplicated data***

```
# check duplicates
print("Number of duplicated data:", data.duplicated().sum())
print("Number of duplicated ID:", data["ID"].duplicated().sum())
```

```
Number of duplicated data: 0
Number of duplicated ID: 0
```

# 1. Data Extraction

## Features & Label

```
label column: Reached.on.Time_Y.N
```

```
numerical columns:
 Index(['Customer_care_calls', 'Customer_rating', 'Cost_of_the_Product',
        'Prior_purchases', 'Discount_offered', 'Weight_in_gms'],
       dtype='object')
```

```
categorical columns:
 Index(['Warehouse_block', 'Mode_of_Shipment', 'Product_importance', 'Gender'], dtype='object')
```

**Column 'ID' is dropped** due to unique data

All features, **except 'ID'**, will be used to see the correlation and insights

# 2. Exploratory Data Analysis (EDA)

## Descriptive Analysis

```
numericals.describe()
```

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

```
categoricals.describe()
```

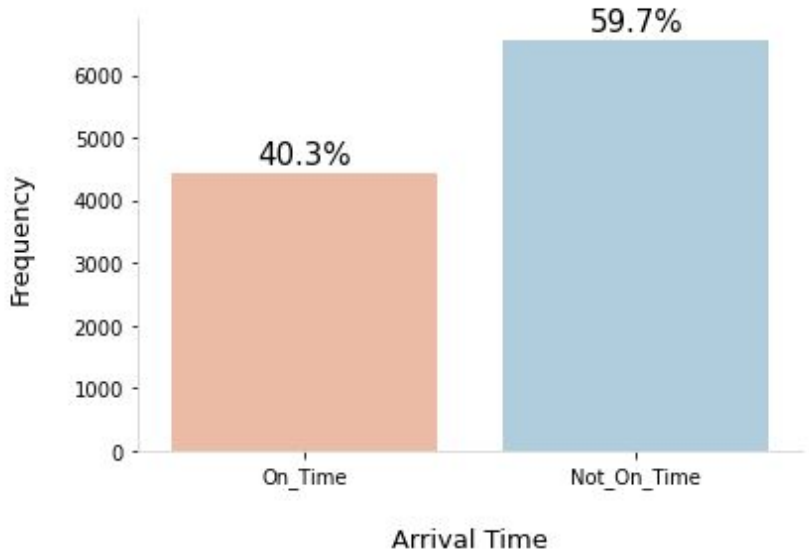
| | Warehouse_block | Mode_of_Shipment | Product_importance | Gender |
|---|---|---|---|---|
| **count** | 10999 | 10999 | 10999 | 10999 |
| **unique** | 5 | 3 | 3 | 2 |
| **top** | F | Ship | low | F |
| **freq** | 3666 | 7462 | 5297 | 5545 |

There is a significant difference between **mean and median** of **'Discount_offered' and 'Weight_in_gms'** which indicates **skewed distribution**

# 2. Exploratory Data Analysis (EDA)

## Univarate Analysis (Countplot)

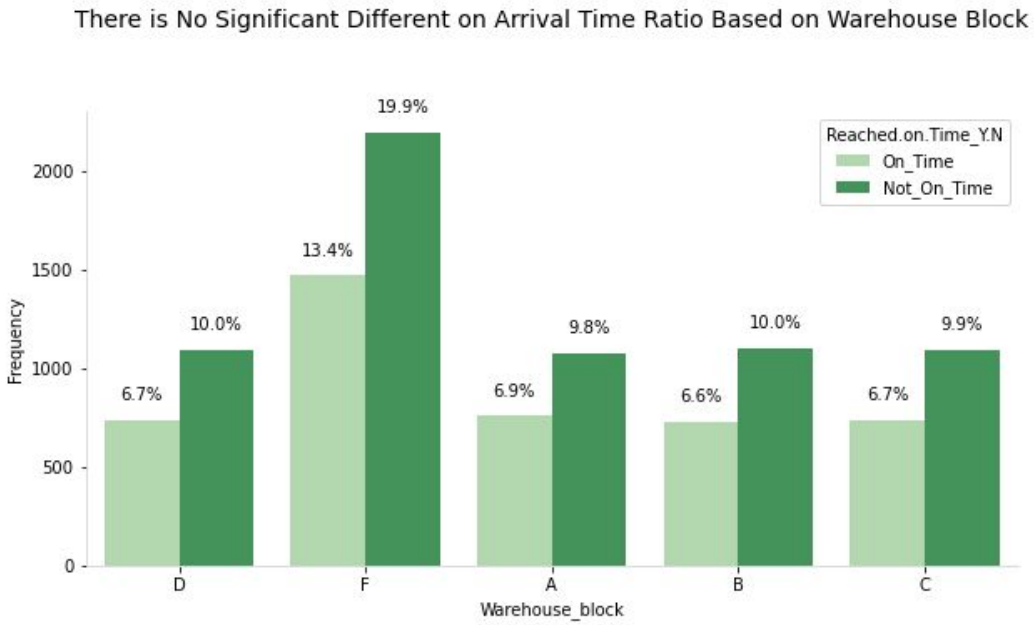
Number of Product Orders that on Time and Delayed are almost Balance




Arrival time categories that are 'On Time' and 'Not On Time' have an almost balanced order frequency

# 2. Exploratory Data Analysis (EDA)

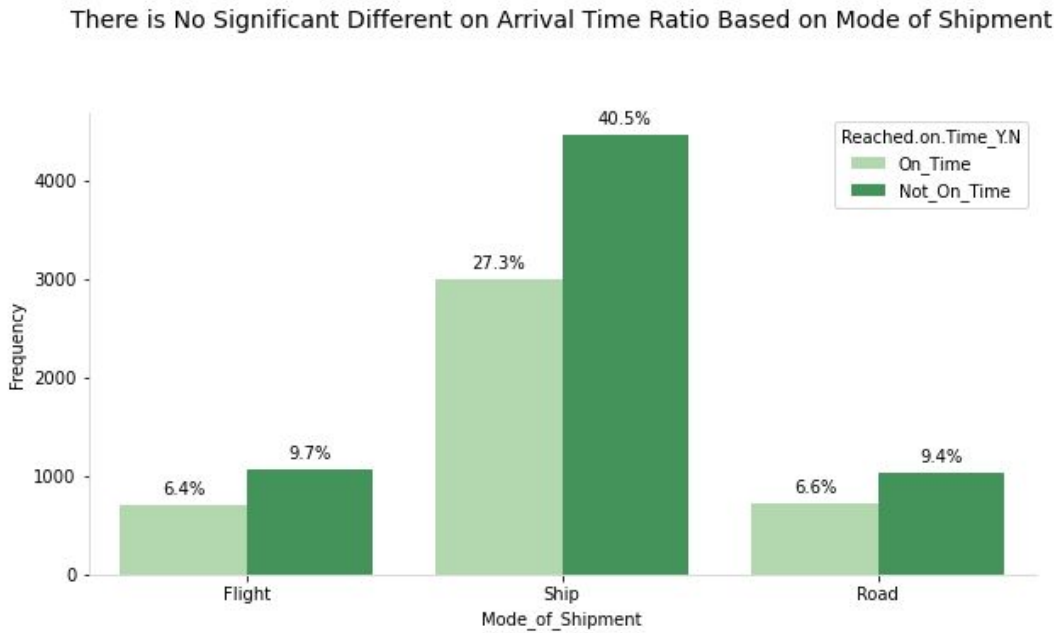
## Univarate Analysis (Countplot)




- *ratio D: 1.49*
- *ratio F: 1.48*
- *ratio A: 1.42*
- *ratio B: 1.52*
- *ratio C: 1.47*

# 2. Exploratory Data Analysis (EDA)

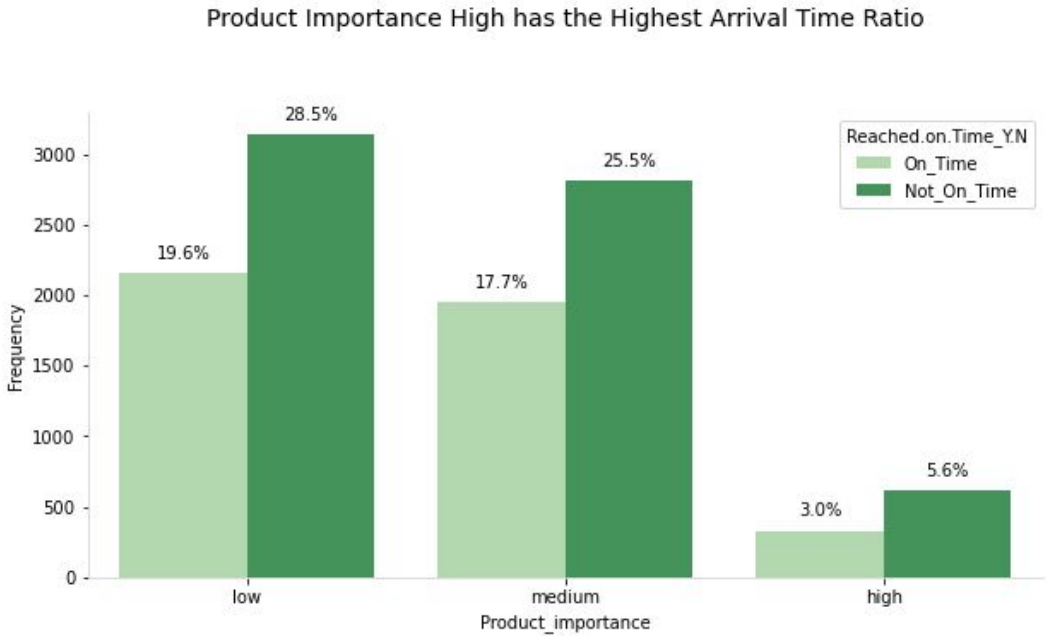
## Univarate Analysis (Countplot)




- *ratio Flight: 1.52*
- *ratio Ship: 1.48*
- *ratio Road: 1.42*

# 2. Exploratory Data Analysis (EDA)

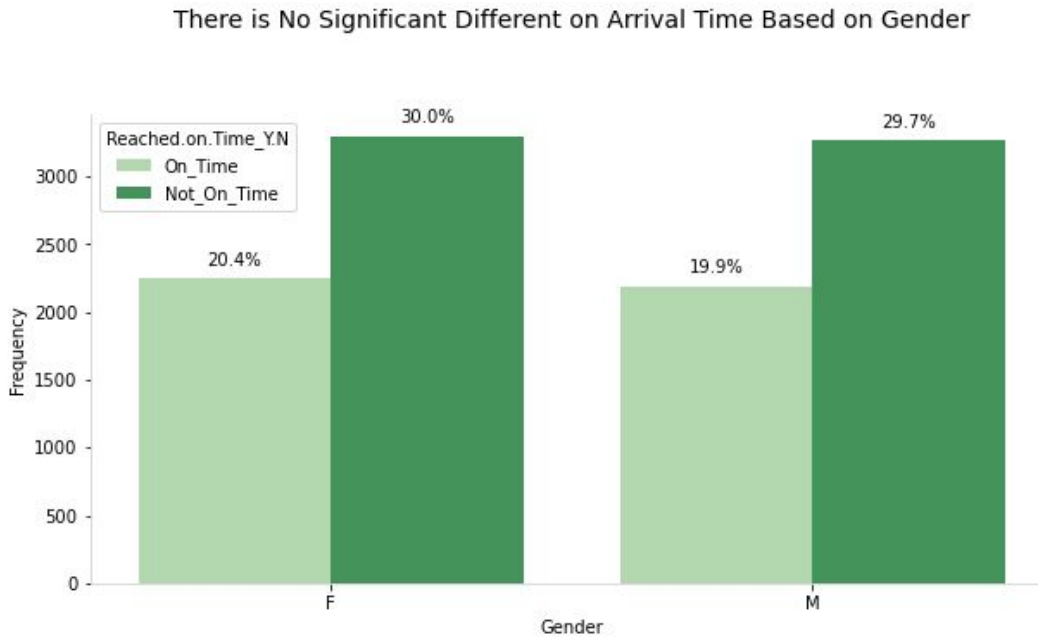
## Univarate Analysis (Countplot)




- *ratio low: 1.45*
- *ratio medium: 1.44*
- *ratio high: 1.87*

# 2. Exploratory Data Analysis (EDA)

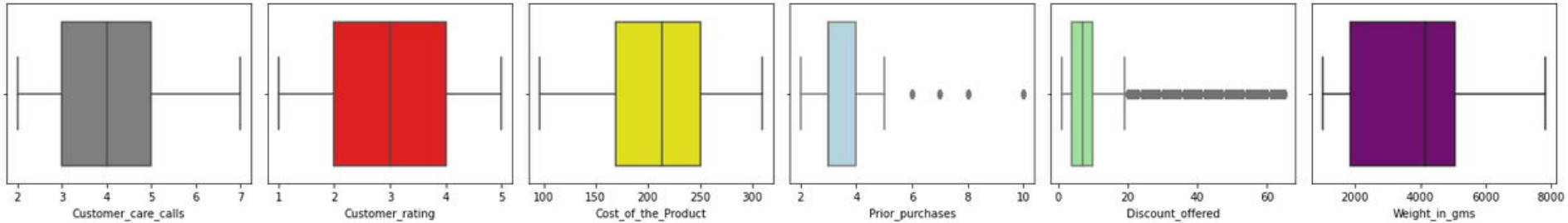
## Univarate Analysis (Countplot)




- *ratio female: 1.47*
- *ratio male: 1.49*

# 2. Exploratory Data Analysis (EDA)

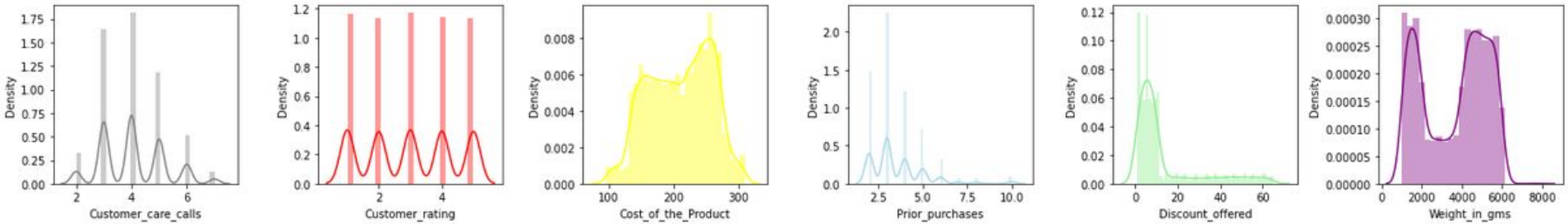
## Univarate Analysis (Boxplots)




- The boxplot graphs above show that 'Prior_purchase' and 'Discount_offered' have a distribution affected by outliers shown on the graph. The outliers are piled up at the right end
- Weight_in_gms has an asymmetric distribution shown by median line that is not in the middle of interquartile range. However, there are no outliers on the graph. It may be affected by the number of modes that is more than 1 and imbalanced. The number of modes indicates there are different groups of Weight
- 'Customer_care_calls','Customer_rating','Cost_of_the_Product' tend to have a normal distribution shown by median line in the midle of IQR

# 2. Exploratory Data Analysis (EDA)

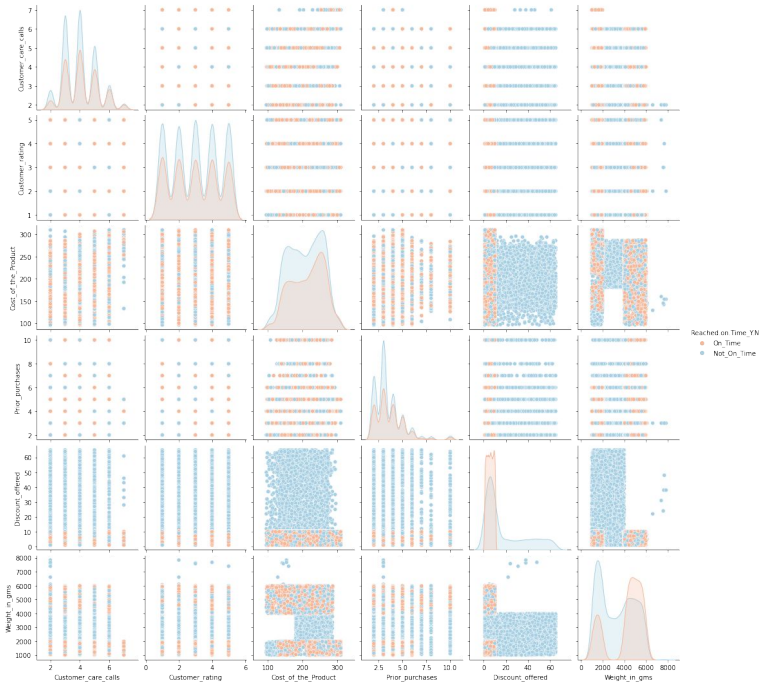
## Univarate Analysis (Boxplots)




- The distribution plots above show that 'Prior_purchase' and 'Discount_offered' have a skewed right (positively skewed) distribution where the data are piled up at the left end. There is a very small portion of the data collected on right side (outliers). It causes the mean > median
- Weight_in_gms has a bimodal distribution. It is affected by 2 imbalanced modes/peaks. It also indicates there are 2 different types of Weight
- 'Customer_care_calls','Customer_rating','Cost_of_the_Product' tend to have a symmetric or normal distribution shown by nearly identical if folded in half at the center point of the distribution
- 'Customer_rating' has a uniform distribution, the probabilities are exactly the same at each point, so the distribution is basically a straight line

# 2. Exploratory Data Analysis (EDA)

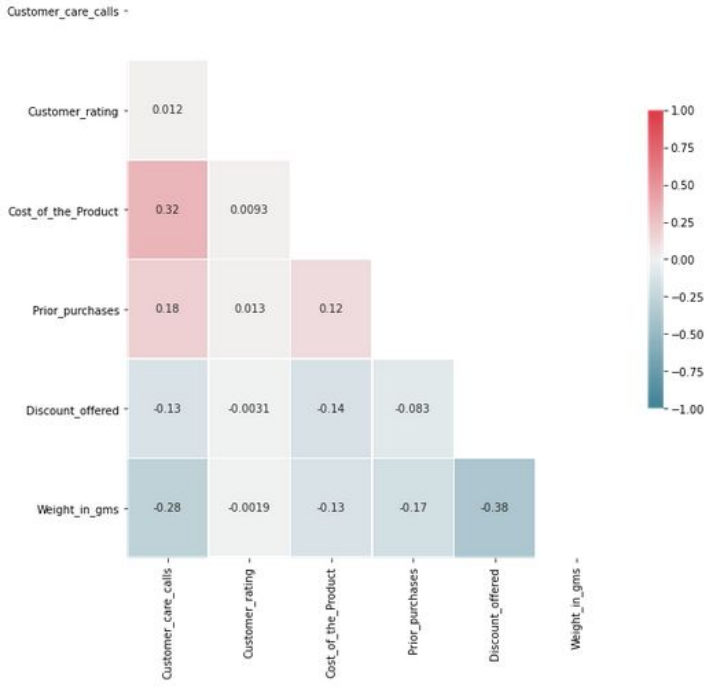
## Multivariate Analysis (Pairplot + Hue)




- Product orders that on time tend to have the lowest 'Discount offered' and tend to have a highest and lowest 'Weight_in_gms'
- The on time and not on time categories on feature correlations between 'Discount_offered' and the other features, as well as feature correlation between 'Weight_in_gms' and the other features, tend to have well separated indicating a good combination of features

# 2. Exploratory Data Analysis (EDA)

## Multivariate Analysis (Correlation Heatmap)




- There are no redundant features, no strong correlated features
- 'Customer_care_calls' are positively correlated with 'Cost_of_the_Product' (r=0.32)
- There are negative correlation between 'Customer_care_calls' and 'Weight_in_gms' (r=-0.28), 'Discount_offered' and 'Weight_in_gms' (r=-0.38)

# Model Evaluation

| | Accuracy | Precision | Recall | F1 Score | AUC Score |
|---|---|---|---|---|---|
| Logistic Regression | 0.64 | 0.69 | 0.70 | 0.70 | 0.62 |
| Logistic Regression GridSearch | 0.59 | 0.59 | 1.00 | 0.74 | 0.50 |
| K-nearest Neighbor | 0.59 | 0.65 | 0.67 | 0.66 | 0.57 |
| K-nearest Neighbor RandomSearch | 0.61 | 0.64 | 0.80 | 0.71 | 0.57 |
| K-nearest Neighbor GridSearch | 0.61 | 0.63 | **0.83** | 0.72 | 0.56 |
| Decision Tree | 0.64 | 0.70 | 0.71 | 0.70 | 0.63 |
| Decision Tree GridSearch | 0.59 | 0.59 | 1.00 | 0.74 | 0.50 |

# Model Evaluation

| | Accuracy | Precision | Recall | F1 Score | AUC Score |
|---|---|---|---|---|---|
| Random Forest | 0.66 | 0.75 | 0.65 | 0.69 | 0.66 |
| Random Forest RandomSearch | 0.66 | 0.75 | 0.65 | 0.70 | 0.67 |
| AdaBoost | 0.68 | 0.77 | 0.65 | 0.71 | 0.69 |
| AdaBoost RandomSearch | 0.68 | 0.78 | 0.64 | 0.70 | 0.69 |
| XGBoost | 0.66 | 0.75 | 0.66 | 0.70 | 0.67 |
| XGBoost RandomSearch | 0.59 | 0.59 | 1.00 | 0.74 | 0.50 |
| Voting Classifier | 0.65 | 0.70 | 0.71 | 0.70 | 0.63 |

# Potential Gross Profit
## (Calculation)

| | |
|---|---|
| Percentage of churned customers reduce (a) | = 8,466% |
| Total revenue from *express shipping* (b) | = ~42 million |
| Total revenue from product + *reguler shipping* (c) | = ~35 billion |
| Potensial Gross Profit | = (b + c) x a |
| | = **~3 billion** |

The prediction of potential profit earned by **Electroux** is

**~3 billion rupiah**