Rakamin Academy

Grow-Ject: Tracking Ecommerce Shipping

Dokumen Laporan Final Project

Latar Belakang Masalah

Link to dataset

- (Role) Sebagai tim data science dari suatu perusahaan e-commerce yang bernama Grow-Ject, kami diminta untuk memberikan solusi bisnis untuk mengatasi masalah bisnis yang ada dengan tujuan meningkatkan performa bisnis berdasarkan data yang tersedia
- (Problem Statement) Sebuah studi mengatakan bahwa 69% customer kemungkinan tidak kembali lagi jika barang telat sampai (source). ~60% product shipment yang dikirim oleh Grow-Ject merupakan *late delivery*. Bisnis prihatin bahwa *late delivery rate* yang cenderung tinggi mempengaruhi *customer retention rate*

Latar Belakang Masalah

- 69% of consumers “are much less or less likely to shop with a retailer in the future if an item they purchased is not delivered within two days of the date promised.”
- 17% of respondents will stop shopping with a retailer after receiving a late delivery one time.
- 55% of respondents will stop shopping with a retailer after receiving a late delivery two to three times.

Source: Hollingsworth

Latar Belakang Masalah

- (Goals) Meningkatkan *customer retention rate*
- (Objectives)
  - Membuat model machine learning yang bertujuan untuk memprediksi *late delivery*
  - Menggali data untuk mengetahui penyebab *late delivery*
  - Memberikan treatment yang tepat kepada customer yang shipmentnya diprediksi sebagai *late delivery* untuk me-retain customer (sebagai bagian dari risk control dalam risk management)
- (Business Metrics) *Customer retention rate*

EDA

Untuk memudahkan penggunaan kolom-kolom yang ada, nama kolom ditransformasikan sebagai berikut:

- 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
- late_delivery

Rakamin Academy

EDA

- Terdapat **10999** observasi
- Semua **tipe data** sudah benar
- Tidak ada **null** values
- Tidak ada **duplicated** values

```
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  late_delivery        10999 non-null  int64
```

```
df.duplicated().sum()

0
```

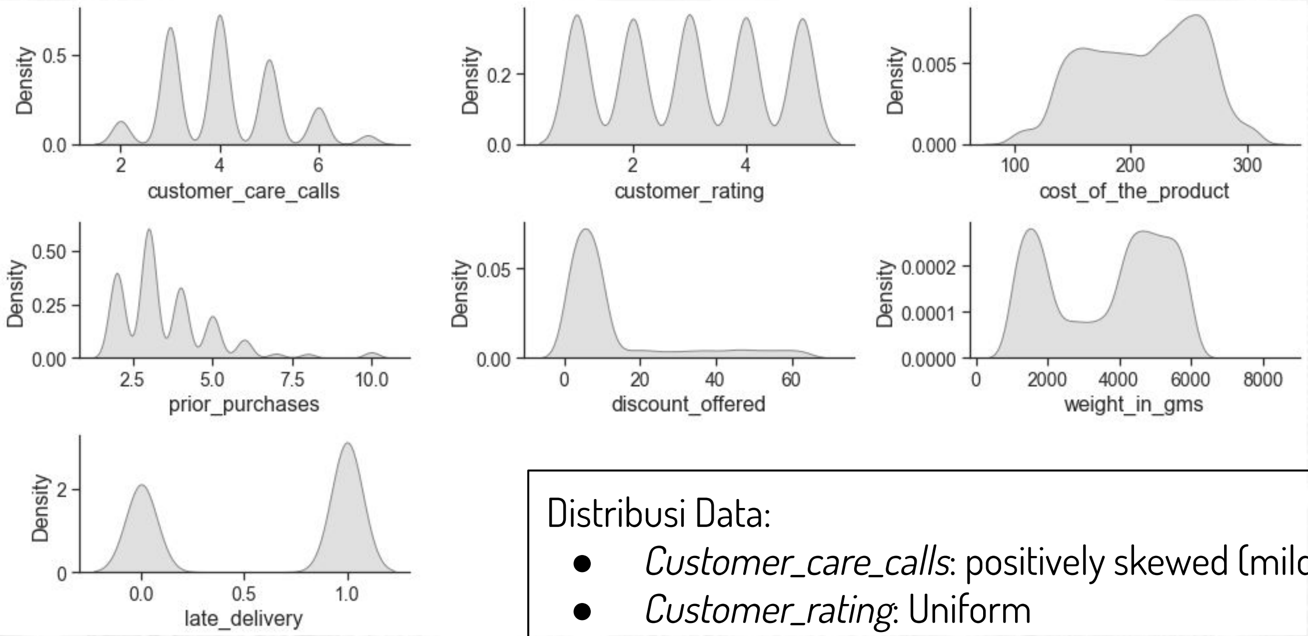
EDA NUMERICAL

	customer_care_calls	customer_rating	cost_of_the_product	prior_purchases	discount_offered	weight_in_gms	late_delivery
count	10999.000000	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	0.596691
std	1.141490	1.413603	48.063272	1.522860	16.205527	1635.377251	0.490584
min	2.000000	1.000000	96.000000	2.000000	1.000000	1001.000000	0.000000
25%	3.000000	2.000000	169.000000	3.000000	4.000000	1839.500000	0.000000
50%	4.000000	3.000000	214.000000	3.000000	7.000000	4149.000000	1.000000
75%	5.000000	4.000000	251.000000	4.000000	10.000000	5050.000000	1.000000
max	7.000000	5.000000	310.000000	10.000000	65.000000	7846.000000	1.000000

Hasil Pengamatan:

- Kolom **customer_care_calls** dan **customer_rating** terlihat distribusinya cukup simetrik (*Mean* dan *Median* tidak berbeda jauh)
- Kolom **prior_purchases** dan **discount_offered** terlihat *Positively Skewed*
- Kolom **late_delivery** bernilai *boolean*

EDA NUMERICAL

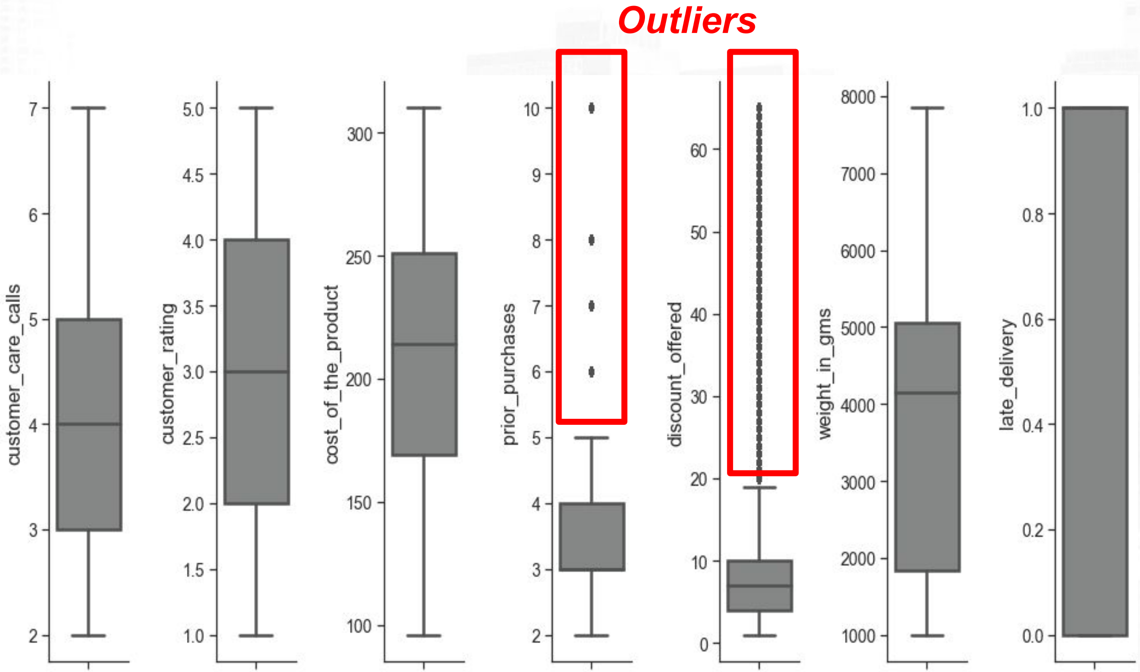


Distribusi Data:

- *Customer_care_calls*: positively skewed (mild)
- *Customer_rating*: Uniform
- *Cost_of_product*: Bimodal
- *Prior_purchases*: Positively Skewed
- *Discount_offered*: Positively Skewed (severe)
- *Weight_in_grams*: U-Shaped / Bimodal

EDA NUMERICAL

Rakamin Academy



Kesimpulan:

1. Outlier paling banyak ditemukan pada parameter ***Discount_offered***, sedangkan pada fitur ***Prior_purchases*** memiliki beberapa outlier yang jelas terlihat
2. ***Weight_in_gms*** memiliki bentuk bimodal distribution

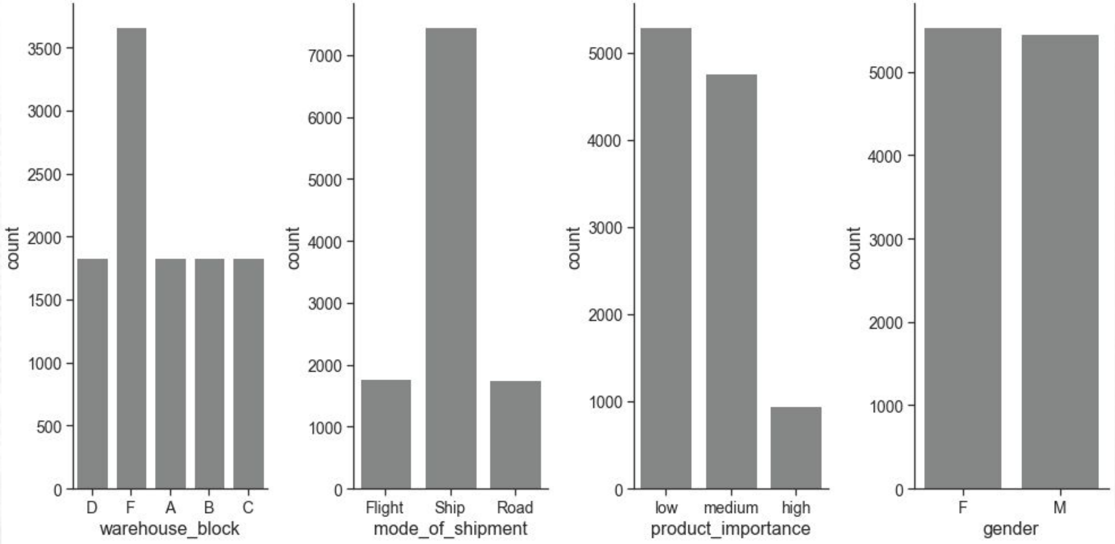
EDA CATEGORICAL DATA

```
warehouse_block :
F    3666
D    1834
A    1833
B    1833
C    1833
Name: warehouse_block, dtype: int64

mode_of_shipment :
Ship      7462
Flight    1777
Road      1760
Name: mode_of_shipment, dtype: int64

product_importance :
low       5297
medium    4754
high       948
Name: product_importance, dtype: int64

gender :
F    5545
M    5454
Name: gender, dtype: int64
```

Hasil pengamatan :

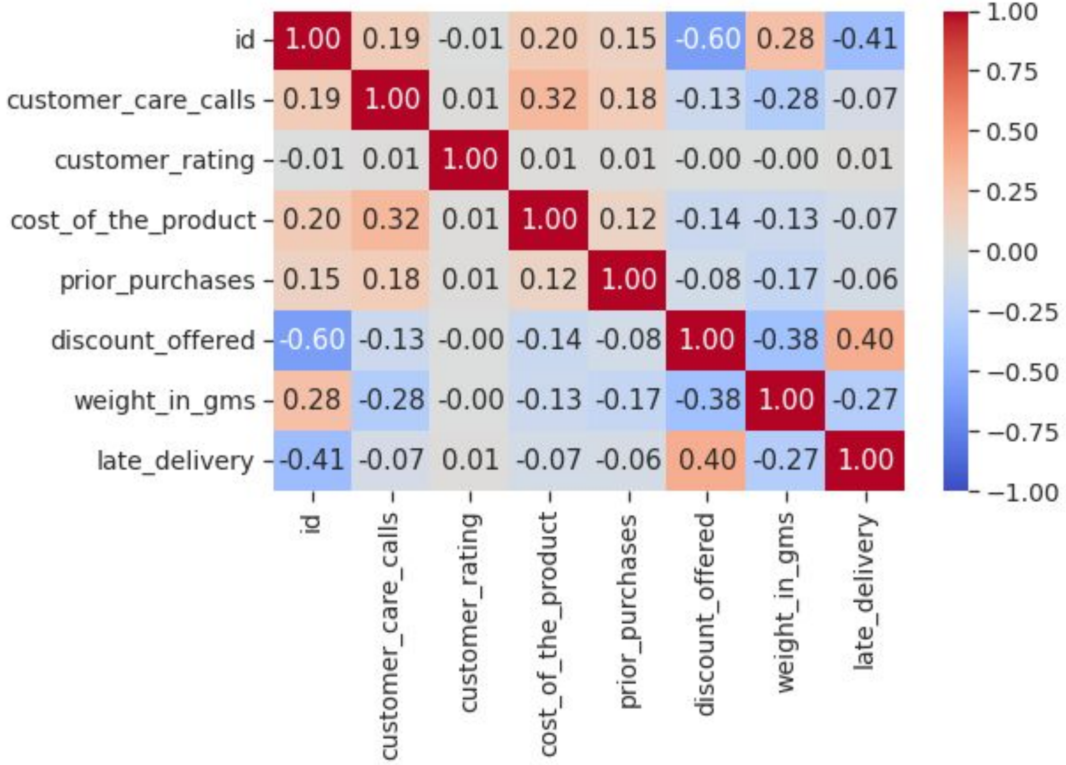
- Kebanyakan pengiriman dilakukan dari gudang **F**
- Metode pengiriman di dominasi oleh pengiriman menggunakan **Ship**
- Kebanyakan barang yang dikirim itu prioritasnya **Low**

EDA CATEGORICAL DATA



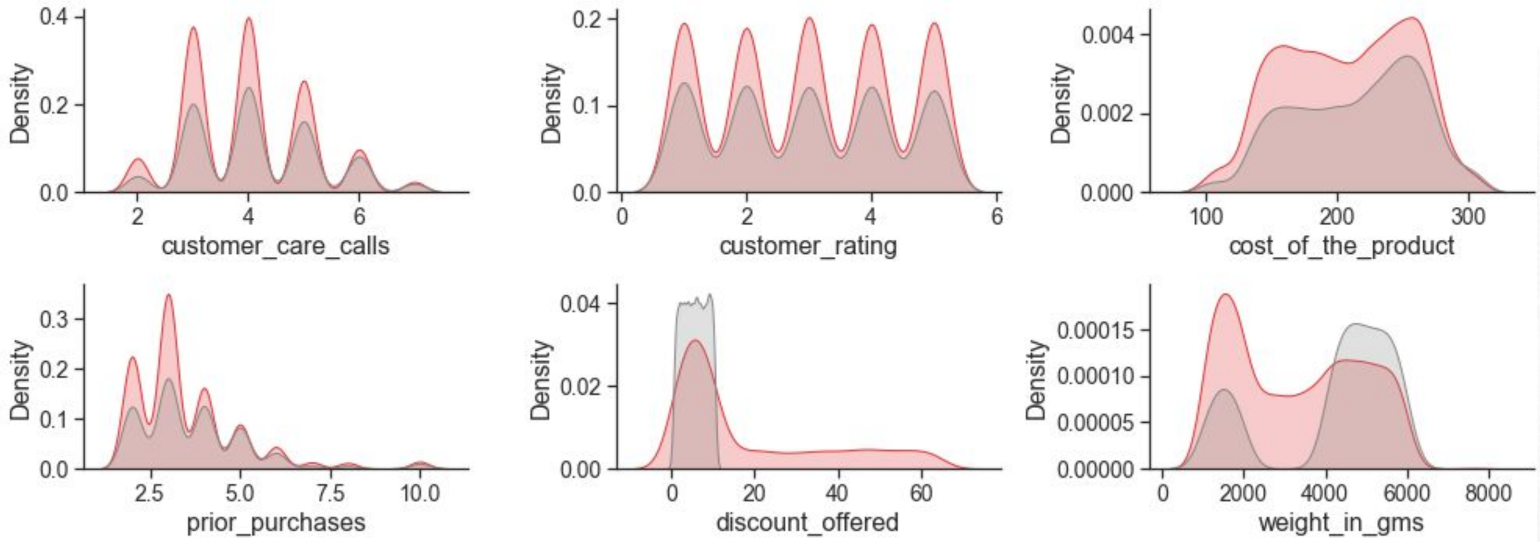
- \`*Warehouse_block*\` F memiliki shipment paling banyak
- Ship adalah \`*mode_of_shipment*\` yang paling sering digunakan
- Pada \`*product_mportance*\`, low adalah yang paling banyak, lalu medium, dan baru high, dengan jumlah yang sangat sedikit jika dibandingkan dengan low dan medium lainnya
- Secara \`*gender*\`, tidak ada perbedaan jumlah shipment

MULTIVARIATE ANALYSIS



1. \`**discount_offered**\` memiliki korelasi positif dengan target \`late delivery\` yaitu sebesar 0.40
2. \`**weight_in_gms**\` memiliki korelasi negatif dengan target \`late delivery\` yaitu sebesar -0.27

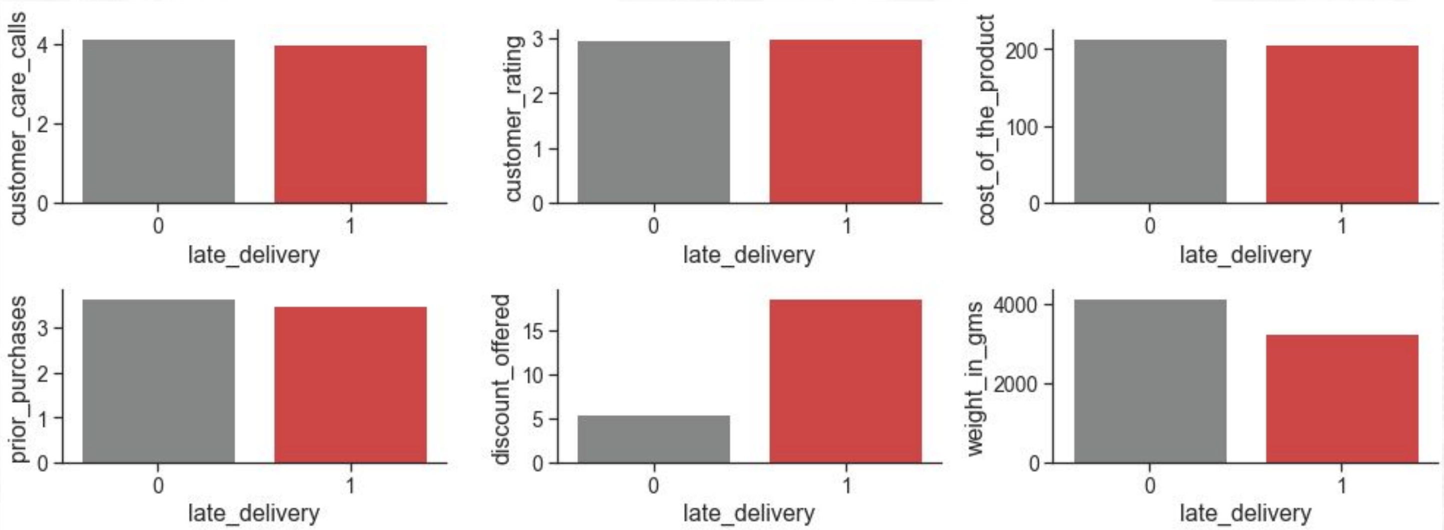
MULTIVARIATE ANALYSIS



Melihat distribusi berdasarkan **late delivery** dan **on-time delivery:**

- Shipment dengan `**cost_of_the_product**` pada sekitar 150 cenderung untuk late dibanding on time
- `**discount_offered**` lebih tinggi untuk late delivery
- `**weight_in_gms**` yang ringan cenderung untuk telat dibanding yang berat

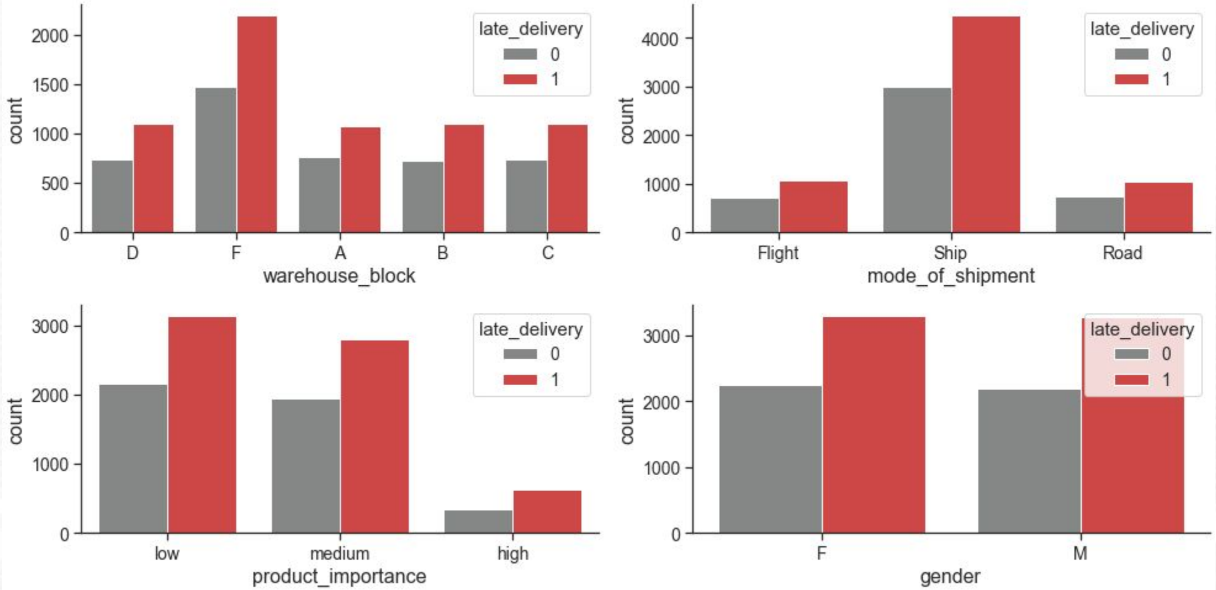
MULTIVARIATE ANALYSIS



Jika kolom numerikal dirata-ratakan berdasarkan **late delivery** dan **on-time delivery:**

- Perbedaan rata-rata terlihat paling jelas pada `**discount_offered**` dan `**weight_in_gms**`
- Terlihat perbedaan yang sangat kecil pada `**cost_of_the_product**`, `**prior_purchases**`, dan `**customer_care_calls**`

MULTIVARIATE ANALYSIS

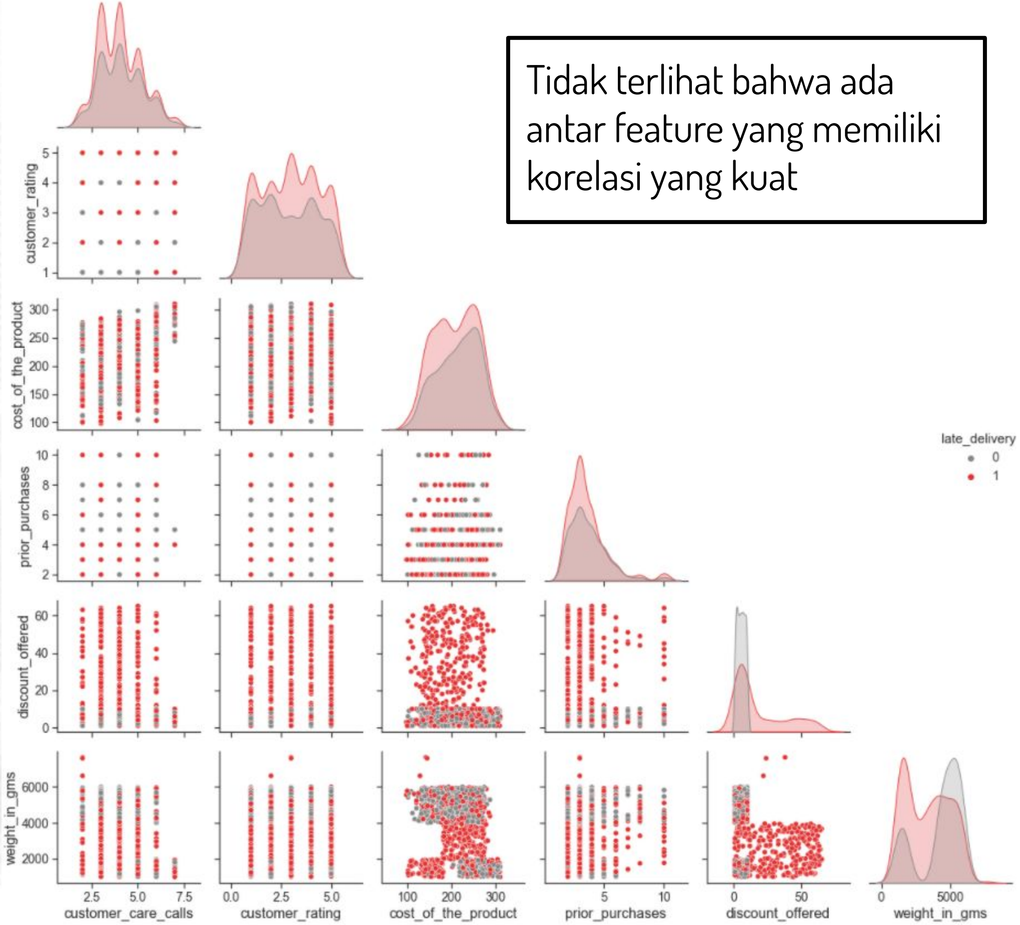


Jika melakukan countplot kolom kategorikal berdasarkan **late delivery** dan **on-time delivery**:

- **warehouse_block** F memiliki late delivery paling banyak diantara semua warehouse block, namun rasio late delivery semua block kurang lebih sama
- Ship memiliki late delivery paling banyak dari semua **mode_of_shipment**
- Secara rasio, **product_importance** high memiliki rasio late delivery paling tinggi
- Secara **gender**, tidak ada perbedaan pada jumlah late delivery

MULTIVARIATE ANALYSIS

Tidak terlihat bahwa ada antar feature yang memiliki korelasi yang kuat



UJI HIPOTESIS (CUSTOMER CARE CALLS)

Menggunakan metode T-Test

Melihat apakah jumlah Customer Care Calls dipengaruhi oleh Late Delivery dan On Time Delivery

```
late_delivery['Customer_care_calls'].mean()

3.9914673167758647

on_time_delivery['Customer_care_calls'].mean()

4.147655545536519
```

```
print('P-Value :',p_value)
if p_value >= 0.05:
    print('Fail to Reject H0')
else:
    print('Reject H0 and Accept H1')

P-Value : 1.8275351786239753e-12
Reject H0 and Accept H1
```

H0 -> customer care calls late delivery == customer care calls on time delivery
H1 -> customer care calls late delivery != customer care calls on time delivery

UJI HIPOTESIS (CUSTOMER CARE CALLS)

Dari uji Hipotesis diatas, dapat disimpulkan bahwa :

- Customer Care Calls Late Delivery **tidak sama dengan** Customer Care Calls On-Time Delivery dan perbedaannya **statistically significant**
- Dari hasil rata-rata, ternyata yang **On-Time Delivery** itu **lebih banyak** Customer Care Calls nya dibandingkan dengan yang **Late Delivery**

UJI HIPOTESIS (CUSTOMER RATING)

Menggunakan metode T-Test

Melihat apakah jumlah Customer Rating dipengaruhi oleh Late Delivery dan On Time Delivery

```
[62] late_delivery['customer_rating'].mean()

     3.005790035044949

[63] on_time_delivery['customer_rating'].mean()

     2.967989179440938
```

```
print('P-Value :',p_value)
if p_value >= 0.05:
    print('Fail to Reject H0')
else:
    print('Reject H0 and Accept H1')

P-Value : 0.16890489722530824
Fail to Reject H0
```

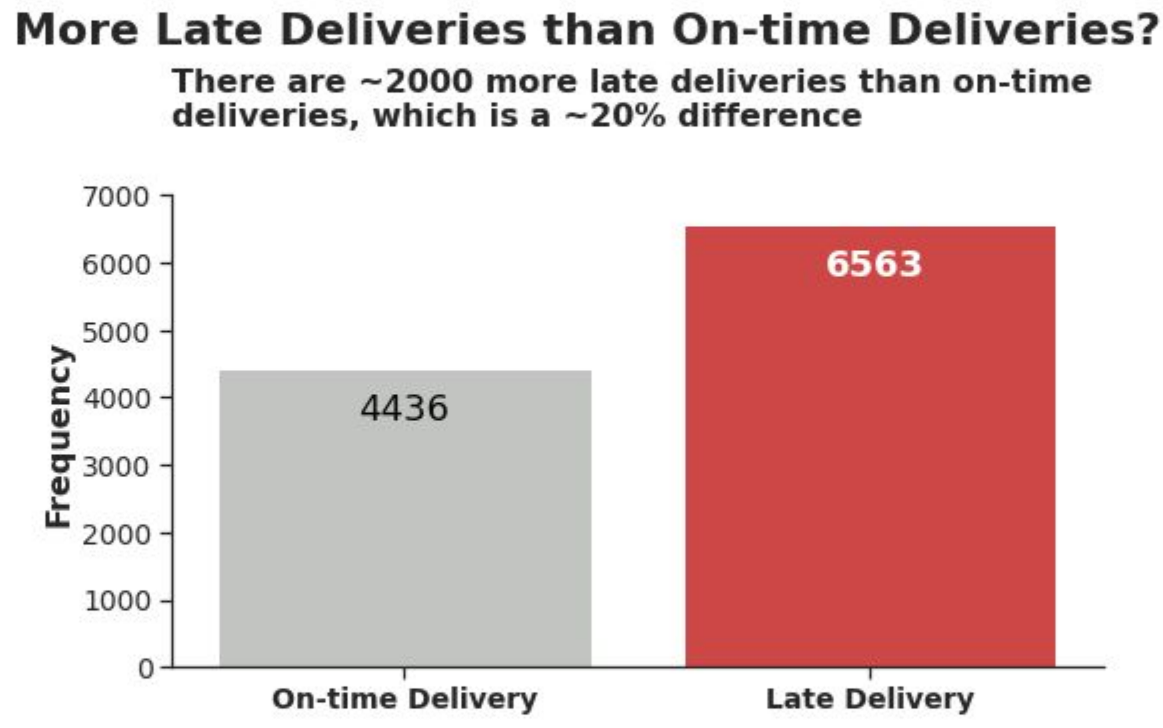
H0 -> customer rating late delivery == customer rating on time delivery
H1 -> customer rating late delivery != customer rating on time delivery

UJI HIPOTESIS (CUSTOMER RATING)

Dari uji Hipotesis diatas, dapat disimpulkan bahwa :

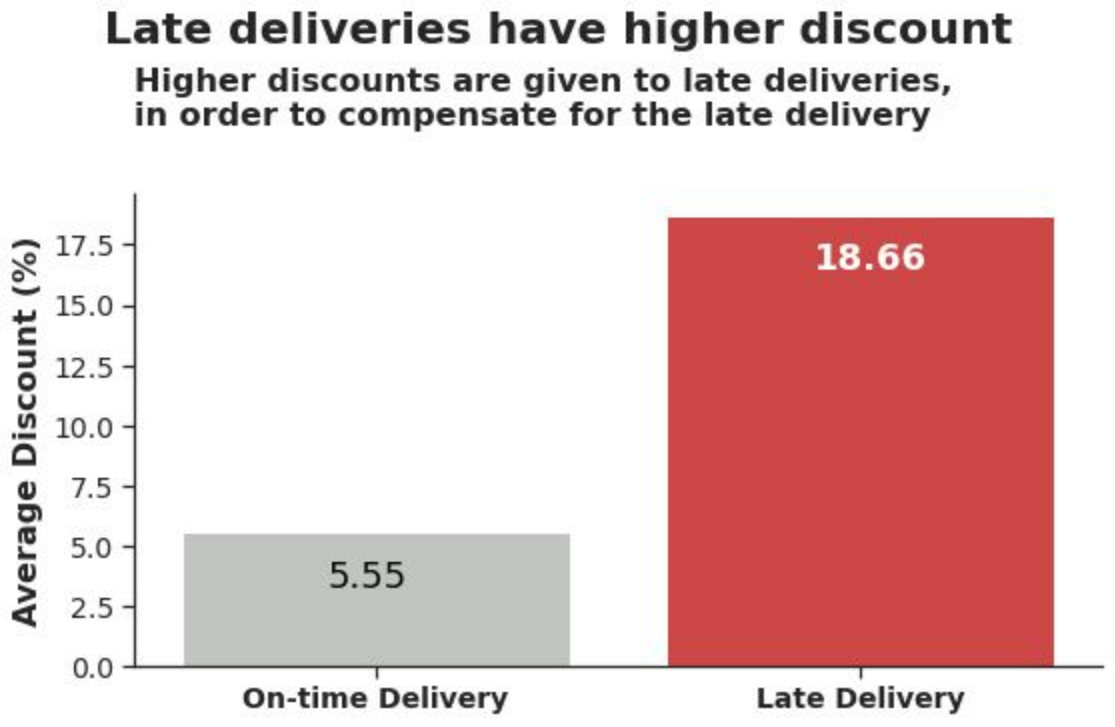
- Customer Rating Late Delivery **sama dengan** Customer Rating On-Time Delivery
- Jadi, customer rating tidak bisa menjadi 'measure of success' untuk on-time delivery rate

Insights



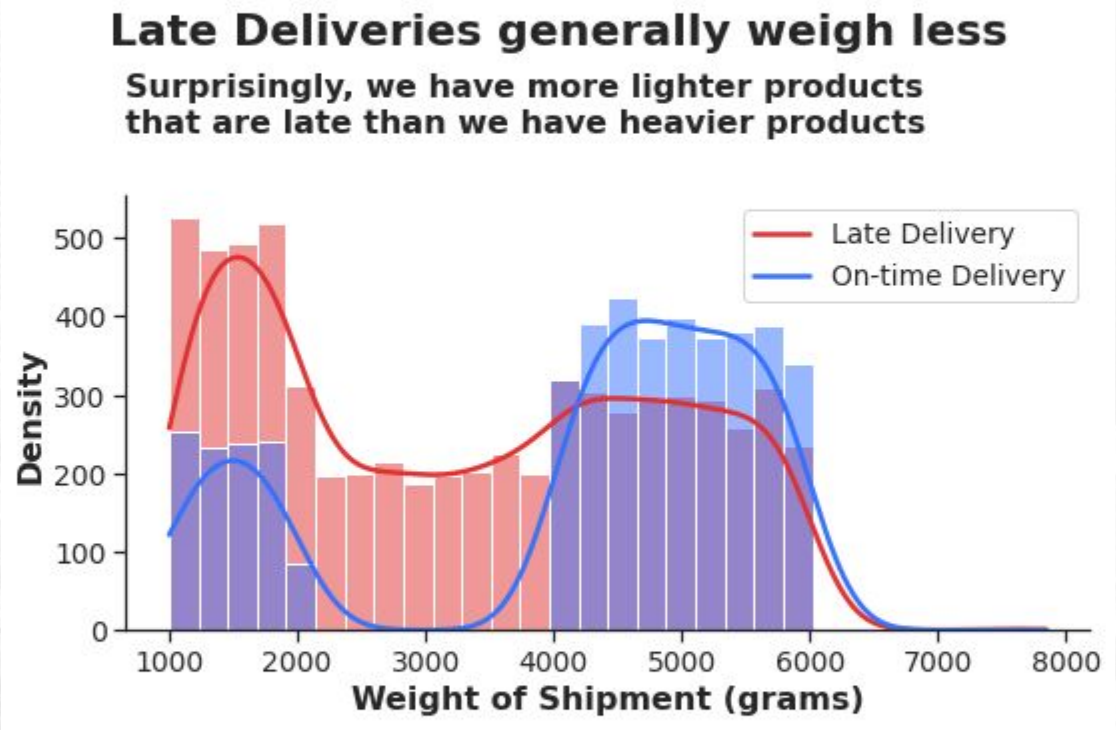
“Six out of 10 deliveries are likely to be delivered late”

Insights



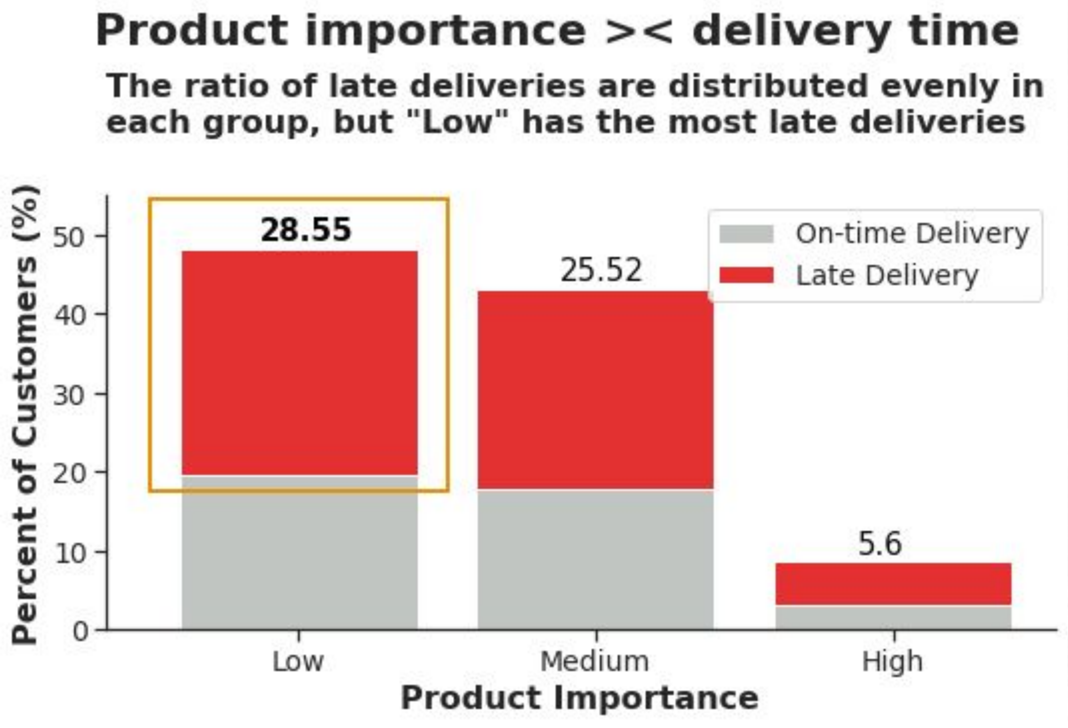
Late deliveries have higher discount
Higher discounts are given to late deliveries, in order to compensate for the late delivery
Average Discount (%)
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
5.55
18.66
On-time Delivery
Late Delivery

Insights



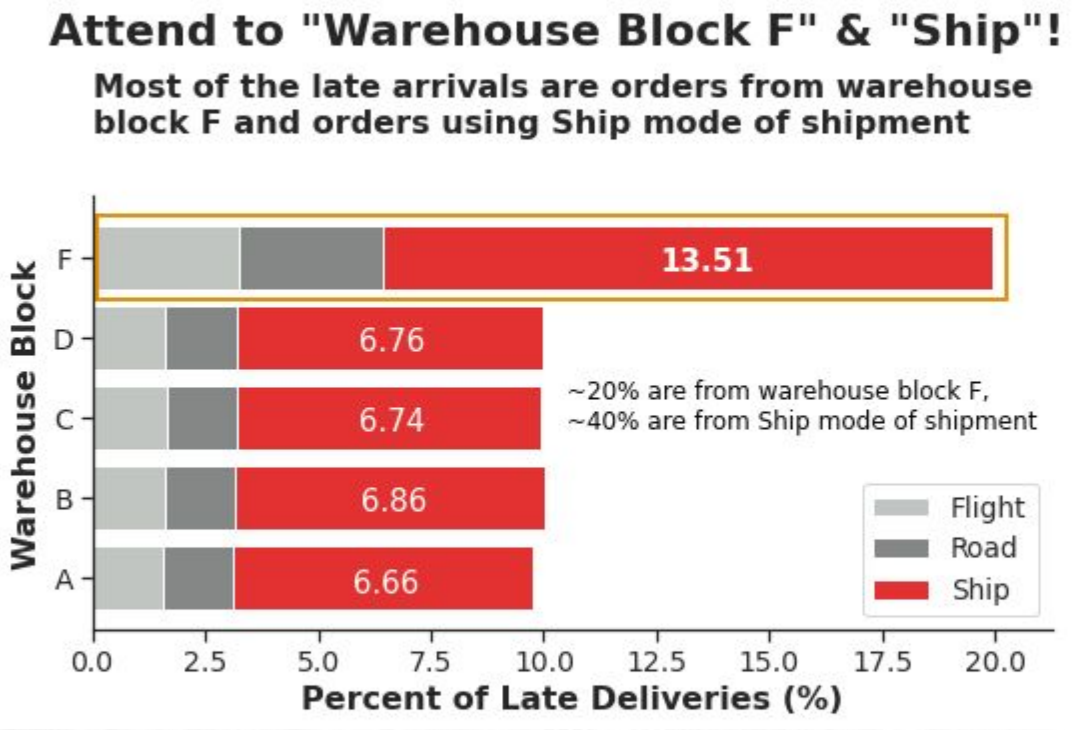
Late Deliveries generally weigh less
Surprisingly, we have more lighter products that are late than we have heavier products
Late Delivery
On-time Delivery
Density
0
100
200
300
400
500
1000
2000
3000
4000
5000
6000
7000
8000
Weight of Shipment (grams)

Insights



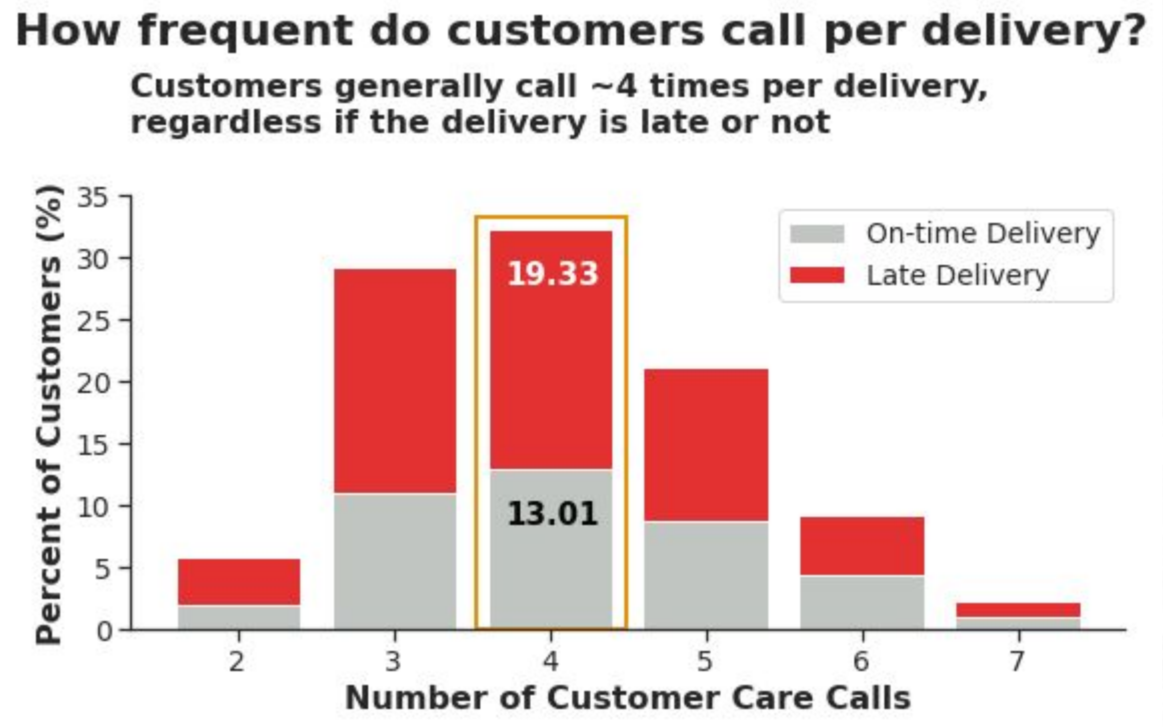
Product importance >< delivery time
The ratio of late deliveries are distributed evenly in each group, but "Low" has the most late deliveries
Percent of Customers (%)
0
10
20
30
40
50
28.55
25.52
5.6
On-time Delivery
Late Delivery
Low
Medium
High
Product Importance

Insights



Attend to "Warehouse Block F" & "Ship"!
Most of the late arrivals are orders from warehouse block F and orders using Ship mode of shipment
Warehouse Block
F
D
C
B
A
13.51
6.76
6.74
6.86
6.66
~20% are from warehouse block F,
~40% are from Ship mode of shipment
Flight
Road
Ship
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Percent of Late Deliveries (%)

Insights



How frequent do customers call per delivery?
Customers generally call ~4 times per delivery,
regardless if the delivery is late or not
Percent of Customers (%)
0
5
10
15
20
25
30
35
19.33
13.01
On-time Delivery
Late Delivery
2
3
4
5
6
7
Number of Customer Care Calls

Insights

```
high_calls = df[df['customer_care_calls']>4]
low_calls = df[df['customer_care_calls']<=4]
```

- high calls have generally higher product cost (+$20) and prior purchases (+0.6)
- high calls have generally less discount offered (-3%) and weight in grams (-850)

- low calls have more late deliveries compared to high calls (6%)

Insights

```
high_rating = df[df['customer_rating']>=4]
low_rating = df[df['customer_rating']<4]
```

- There's nothing different when comparing high rating and low rating, which is an oddity with regards to our project's objectives
- If rating is not affected by any of the features in this dataset, then can be it improved in this case?

Other insights:
- the higher the product importance, generally the heavier
- for late deliveries, more discounts are offered the higher the importance of the product
- the higher the importance of product, the lower the cost of product
- most likely cost of product takes into account of the discount offered

Insights

Issues with dataset:

- Customer rating does not correlate with any other columns → can we still use it as a measure of success in this project?
- Customer care calls does not determine customer rating either so it may not be a good KPI as well
- Is customer ID ordered by their time of order? (there's no order date)
- Prior purchases exist but we can't tell if it's the same customer since all IDs are unique → customer ID might indicate order ID instead
- Simply, there isn't much we can get from such a narrow dataset, so what we can achieve here is very limited :(

Insights

- 69% of consumers "are much less or less likely to shop with a retailer in the future if an item they purchased is not delivered within two days of the date promised." (source)
- 17% of respondents will stop shopping with a retailer after receiving a late delivery one time. (source)
- 55% of respondents will stop shopping with a retailer after receiving a late delivery two to three times. (source)

- Can we switch to another business metric? (eg, based on an article)
    - On-time delivery rate (OTD rate affects customer retention)
    - Revenue (predict revenue from retained customer)
- Example solution: if late delivery is predicted, inform customer (best from text/WA linking to live tracking page) and re-set expectation (new ETA and discount for compensation) → or optimize last-mile delivery

Articles:

- Delivery experience customer behavior
- KPIs

Data Preprocessing

- Tidak ada missing, duplicate, redundant, ataupun invalid data
- Pengecekan outlier menggunakan Z-score dan membuang Z-score yang lebih dari 3

```
Rows before filtering outliers: 10999
Rows after filtering outliers: 10642
```

- Class imbalance:
  40 on-time : 60 late delivery → oversampling SMOTE (frac=0.75)

```
Before Sampling
1     6282
0     4360
```

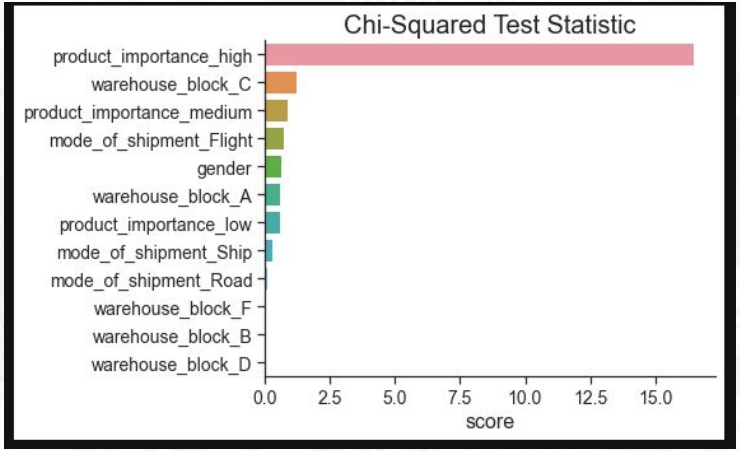
```
After Sampling
1     6282
0     4711
```

Data Preprocessing

- Splitting data menjadi train set dan test set (80:20)
- Transformasi distribusi (np.sqrt untuk weight in gms, prior purchases, discount offered, dan cost of the product)
- Feature scaling (MinMaxScaler)
- Label encoding untuk gender (M = 1 & F = 0)
- OneHotEncoding untuk warehouse_block, mode_of_shipment, product_importance
- Step 5, 6, dan 7 dilakukan kepada test set juga

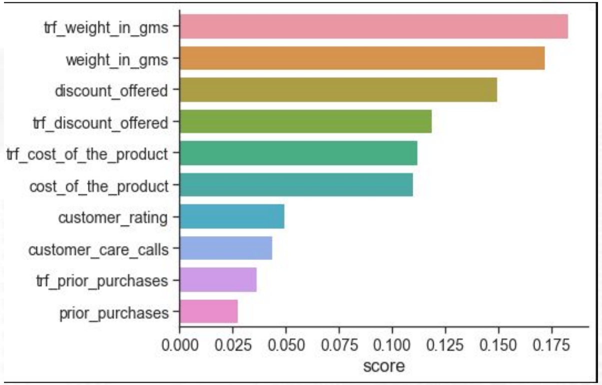
Feature Selection

1. **Chi Squared test**: melihat predictive power dari categorical features. Ternyata hanya **product_importance_high** yang memiliki score tinggi

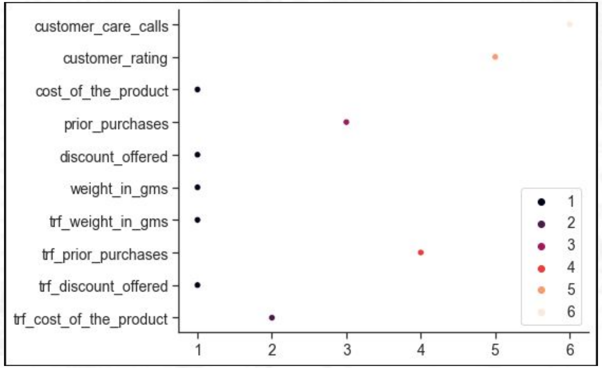


Feature Selection

2. **Feature importances** menggunakan *RandomForestClassifier*:

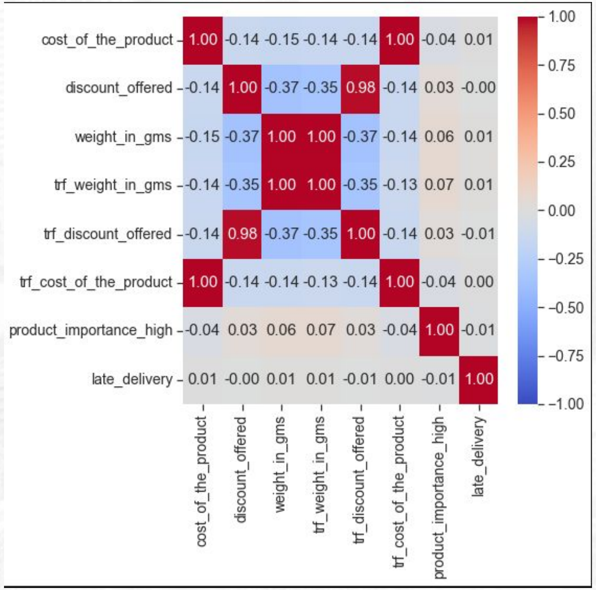


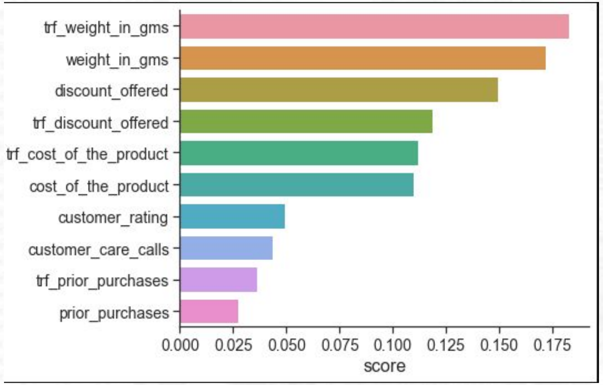
3. **Feature Rankings** menggunakan *Boruta* dari hasil feature importances *RandomForestClassifier*, dan diambil fitur ranking 1 dan 2 saja



Feature Selection

4. **Multicollinearity**: *Pearson correlation matrix* untuk melihat fitur yang redundant, dibuang yang secara feature importance lebih rendah





Feature Selection

4. Melalui tahapan-tahapan feature selection, berikut adalah list fitur yang terseleksi untuk digunakan pada machine learning model:

- *cost_of_the_product*
- *discount_offered*
- *trf_weight_in_gms* (np.sqrt)
- *product_importance_high* (satu-satunya categorical feature yang memiliki chi-squared statistic tinggi

Machine Learning

Tujuan utama dari machine learning model ini adalah memprediksi late delivery. Kami ingin **meminimalisir False Negatives** pada prediksi. Pada kasus ini, prioritas lebih tinggi untuk memprediksi *late delivery* daripada *on-time delivery*. Oleh karena itu, evaluation metric yang tepat adalah **Recall** score.

Recall = TruePositives / (TruePositives + FalseNegatives)

Model selection diawali dengan menggunakan package yang bernama *lazypredict*, yang mencoba prediksi test data dari 29 classifier yang berbeda.

Terlihat bahwa model yang unggul adalah *tree-based, bagging, stacking, dan boosting* models.

Machine Learning

Meneruskan dari *lazypredict*, beberapa model yang unggul diuji kembali.

Berhubung sistem ini adalah hasil dari *sprint* (*Agile*) yang paling awal, kami menetapkan metrics untuk *production grade* sebagai berikut:

- Recall untuk *late delivery* : **0.75**
- Recall untuk *on-time delivery* : **0.5**

Karena pada prosesnya scoring tidak bisa dipilih untuk dua class secara terpisah, maka cross validation dan prediction awal diuji menggunakan metrics yaitu **averaged-micro recall**

Machine Learning

CV Train Scores	
Model	Recall-Micro
DecisionTree	0.6384
ExtraTree	0.6373
AdaBoost	0.6682
Bagging	0.6517
ExtraTrees	0.6442
GradientBoosting	0.6805
RandomForest	0.6537
Stacking	0.6466
Voting	0.6525
HistGradientBoosting	0.6622
XGBoost	0.6542
XGBoostRF	**0.6850**
LightGBM	0.6636

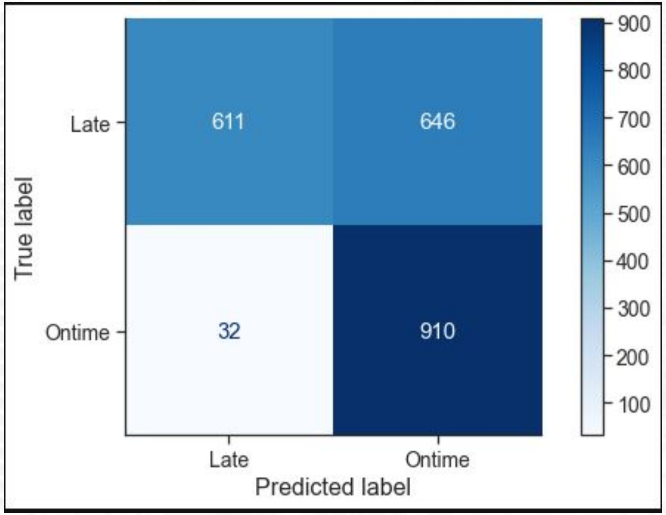
Test Scores	
Model	Recall-Micro
DecisionTree	0.6444
ExtraTree	0.6498
AdaBoost	0.6780
Bagging	0.6662
ExtraTrees	0.6503
GradientBoosting	0.6926
RandomForest	0.6676
Stacking	0.6667
Voting	0.6576
HistGradientBoosting	0.6799
XGBoost	0.6703
XGBoostRF	**0.6917**
LightGBM	0.6917

XGBoostRF merupakan model yang paling unggul (dan paling *viable* untuk hyperparameter tuning)

Machine Learning

Sebelum hyperparameter tuning:

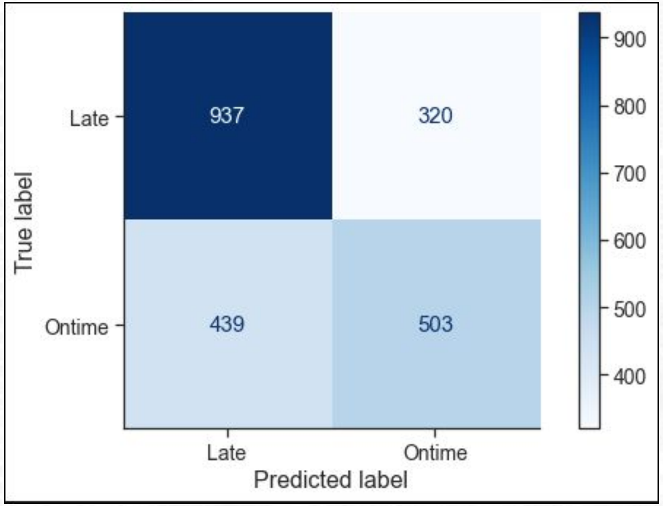
	precision	recall	f1-score	support
Late	0.95	0.49	0.64	1257
Ontime	0.58	0.97	0.73	942
accuracy			0.69	2199
macro avg	0.77	0.73	0.69	2199
weighted avg	0.79	0.69	0.68	2199



Machine Learning

Setelah hyperparameter tuning:

	precision	recall	f1-score	support
Late	0.68	0.75	0.71	1257
Ontime	0.61	0.53	0.57	942
accuracy			0.65	2199
macro avg	0.65	0.64	0.64	2199
weighted avg	0.65	0.65	0.65	2199



Hyperparameters:

- 'booster': 'dart',
- 'eta': 0.025,
- 'gamma': 0,
- 'max_delta_step': 3,
- 'max_depth': 9,
- 'min_child_weight': 0.8,
- 'reg_alpha': 2,
- 'reg_lambda': 0.1,
- 'tree_method': 'hist',
- 'scale_pos_weight': 1.4,
- 'subsample': 0.7,
- 'n_jobs': -1,
- 'random_state': 42,

Business Recommendation