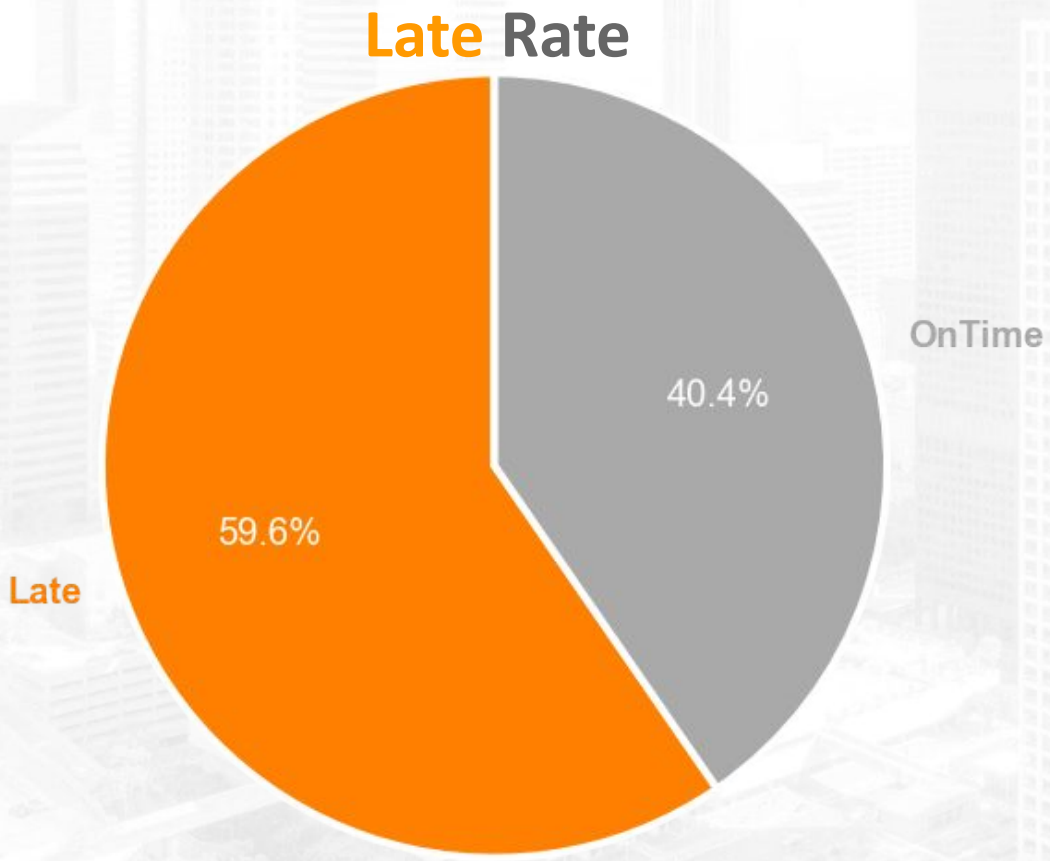
Rakamin Academy

Hexa Avengers

Dokumen Laporan Final Project

- Kevin Usmayadhy Wijaya
- Qistina Muharrifa
- Riel Jeremy Jordan Umboh
- Nabil Abduh Aqil
- Febiya Jomy Pratiwi
- Vicky Clarissa Jennie Damara

Stage 0 - Latar Belakang Masalah



PT. Avengers merupakan perusahaan di bidang *e-commerce* yang sudah memiliki **10.999 transaksi**. Namun terdapat temuan bahwa sebanyak **6.563 (59.6%)** transaksi mengalami keterlambatan hal ini diduga akan mempengaruhi **satisfaction customer**.

Stage 0 - Latar Belakang Masalah

Kami sebagai Tim Data yang terdiri dari:

- Project Leader: Kevin Usmayadhy Wijaya
- Data Analyst: Vicky Clarissa Jennie Damara
- Data Scientist : Nabil Abduh Aqil
- Machine Learning Engineer: Febiya Jomy Pratiwi
- Business Analyst: Qistina Muharrifa & Riel Jeremy Jordan Umboh

Mengidentifikasikan problem, goal, objective, dan business metrics sesuai yang tertera pada tabel.

Problem	Besarnya persentase keterlambatan barang
Goal	Menurunkan persentase keterlambatan barang
Objective	- Membuat model klasifikasi yang bisa memprediksi keterlambatan barang - Mencari faktor-faktor yang mempengaruhi keterlambatan
Business Metrics	Late Rate

Stage 1 - Info Kolom

```
df.info() # data type masing-masing kolom sesuai
```

```
<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
```

- Semua column sudah terisi sehingga tidak perlu dilakukan handling missing value
- Jika dilihat dari columnnya semua sudah memiliki tipe yang sesuai. Nama column **Reach.on.Time_Y.N** diubah agar tidak membingungkan menjadi **Is_Late** karena value 1 merepresentasikan produk terlambat (tidak on time) dan 0 merepresentasikan produk tidak terlambat (on time) sehingga kurang sesuai dengan nama column Reach.on.Time_Y.N.

Stage 1 - Describe Kolom Kategori

	Warehouse_block	Mode_of_Shipment	Product_importance	Gender	Is_Late
count	10999	10999	10999	10999	10999
unique	5	3	3	2	2
top	F	Ship	low	F	True
freq	3666	7462	5297	5545	6563

Berdasarkan unique values, semua variabel sesuai nilainya pada deskripsi dataset, tidak ada kesalahan input. Berdasarkan frequencies dan top frequent dapat terlihat bahwa:

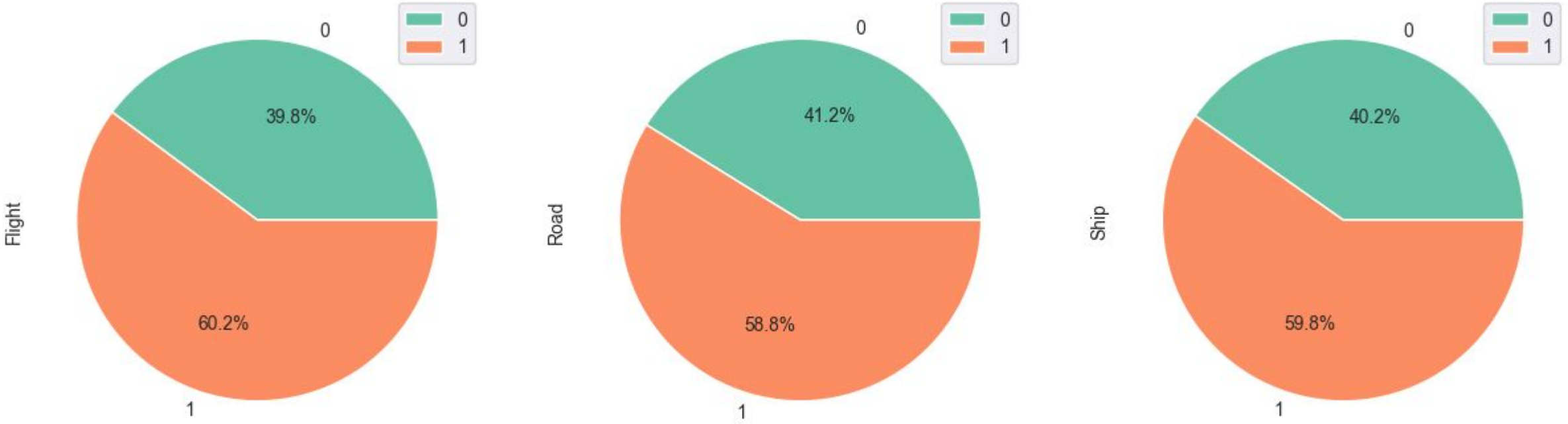
- Pengiriman cenderung terlambat **(Is_Late)** sebanyak **6563**.
- Penyimpanan dominan pada **Warehouse_Block F** sebanyak **3866**.
- **Product_Importance** dengan kategori **Low** sebanyak **5297**.
- Pengiriman **Gender** paling dominan adalah **F (Female)** sebanyak **5545**.
- Barang dikirim menggunakan **Mode of Shipment** terbesar yaitu **Ship** sebanyak **7462**, Hal ini menandakan adanya ketimpangan kelas yang besar dibandingkan jenis pengiriman lainnya.

Stage 1 - Describe Kolom Numerik

	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

- **Customer_care_calls :** customer minimal melakukan telepon sebanyak **2 kali** dengan rata-rata (mean) **4 kali** dan maksimal **7 kali**
- **Customer_rating: :** customer memberikan minimal rating **1** dengan rata-rata (mean) nilai **3** dan maksimal rating **5**
- **Cost_of_the_product :** customer membeli barang dengan harga minimal **96** USD dengan rata-rata (mean) harga **210** USD dan maksimal harga **310** USD.
- **Prior Purchase :** customer melakukan pembelian minimal sebanyak **2 kali** dengan rata-rata pembelian (mean) **3.6 kali** dan maksimal **10 kali**,
- **Discount offered :** customer minimal mendapatkan **1%** diskon dengan rata-rata (mean) **13,37%,** dengan diskon maksimal **65%.**
- **Weights in gms :** Berat barang yang dipesan customer minimal **1001 gram** dengan rata-rata (mean) **3634 gram** dan maksimal berat barang sebesar **7846 gram.**

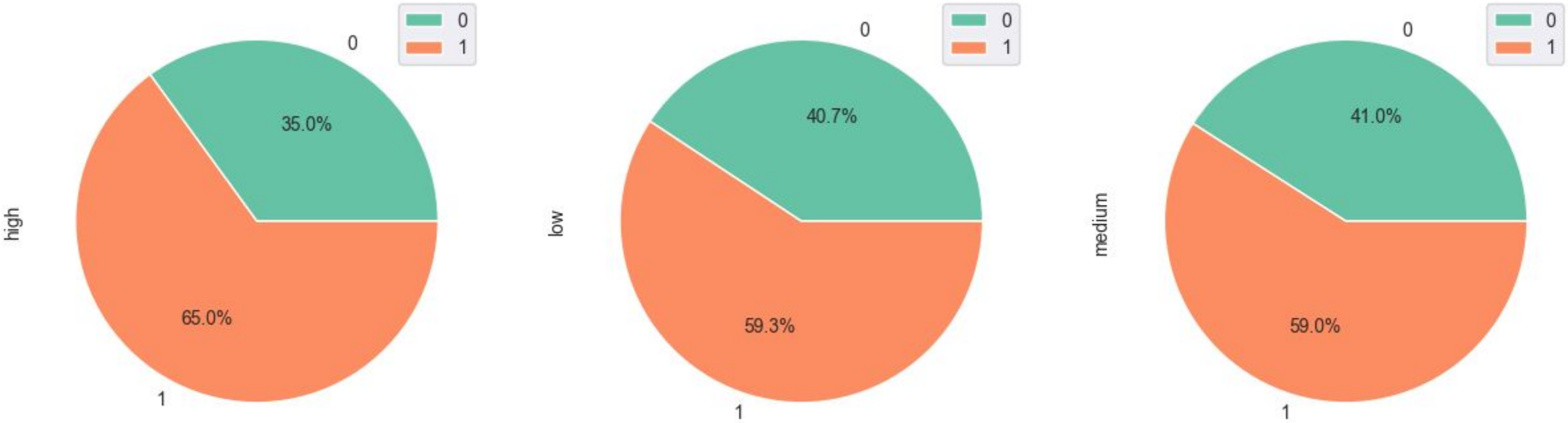
Stage 1 - Mode of Shipment



Berdasarkan pie plot di atas, dapat disimpulkan bahwa:

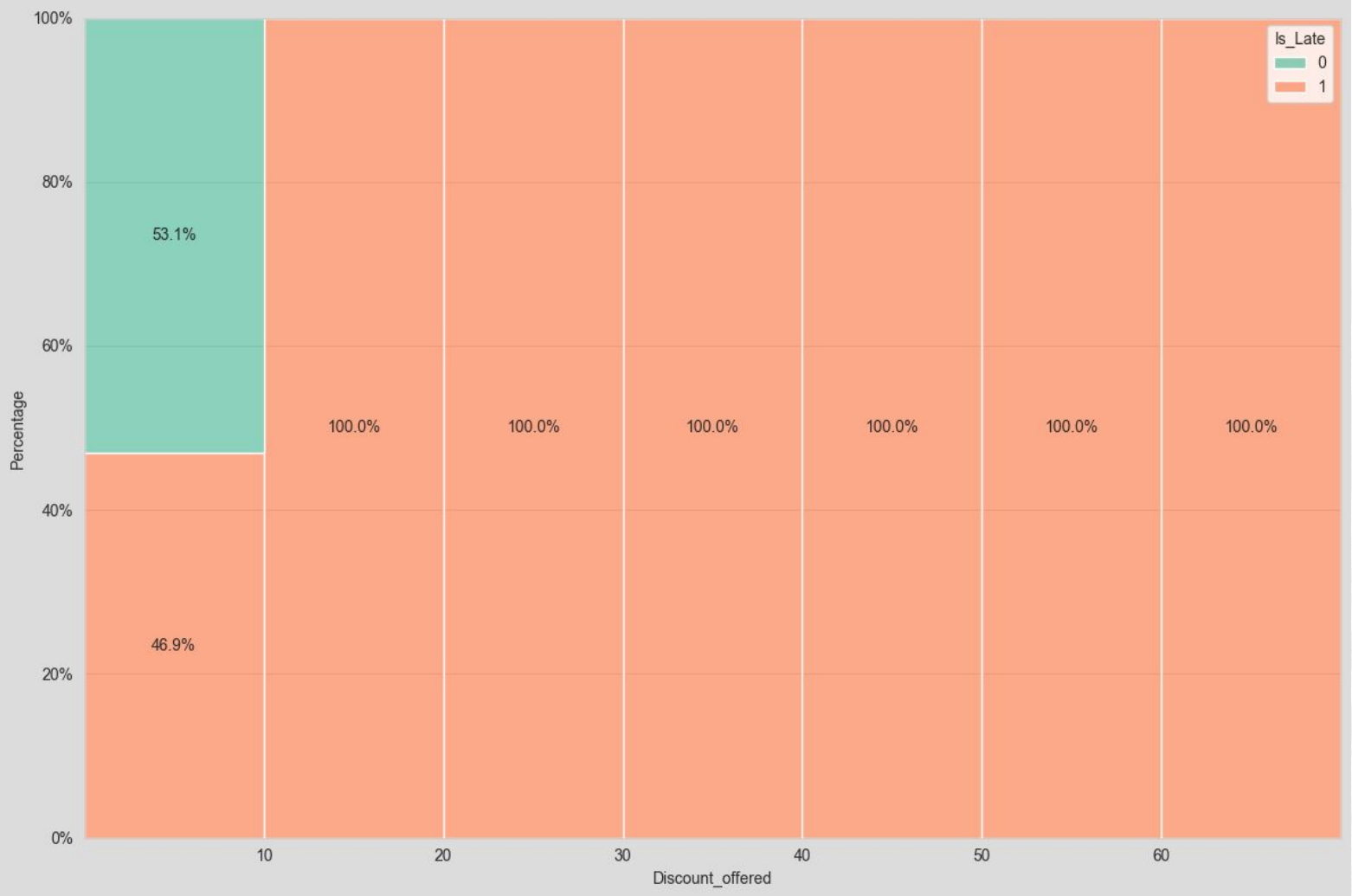
- Barang dengan mode shipment **Flight** memiliki persentase **keterlambatan tertinggi** dibandingkan mode shipment lainnya.
- Barang dengan mode shipment **Road** memiliki persentase **keterlambatan terkecil** dibandingkan kedua mode shipment lainnya.
- Namun persentase keterlambatannya tidak berbeda jauh.

Stage 1 - Product Importance



Baik barang dengan product importance high, medium, dan low tetap mengalami keterlambatan yang relatif besar.

Stage 1 - Discount Problem



Insight

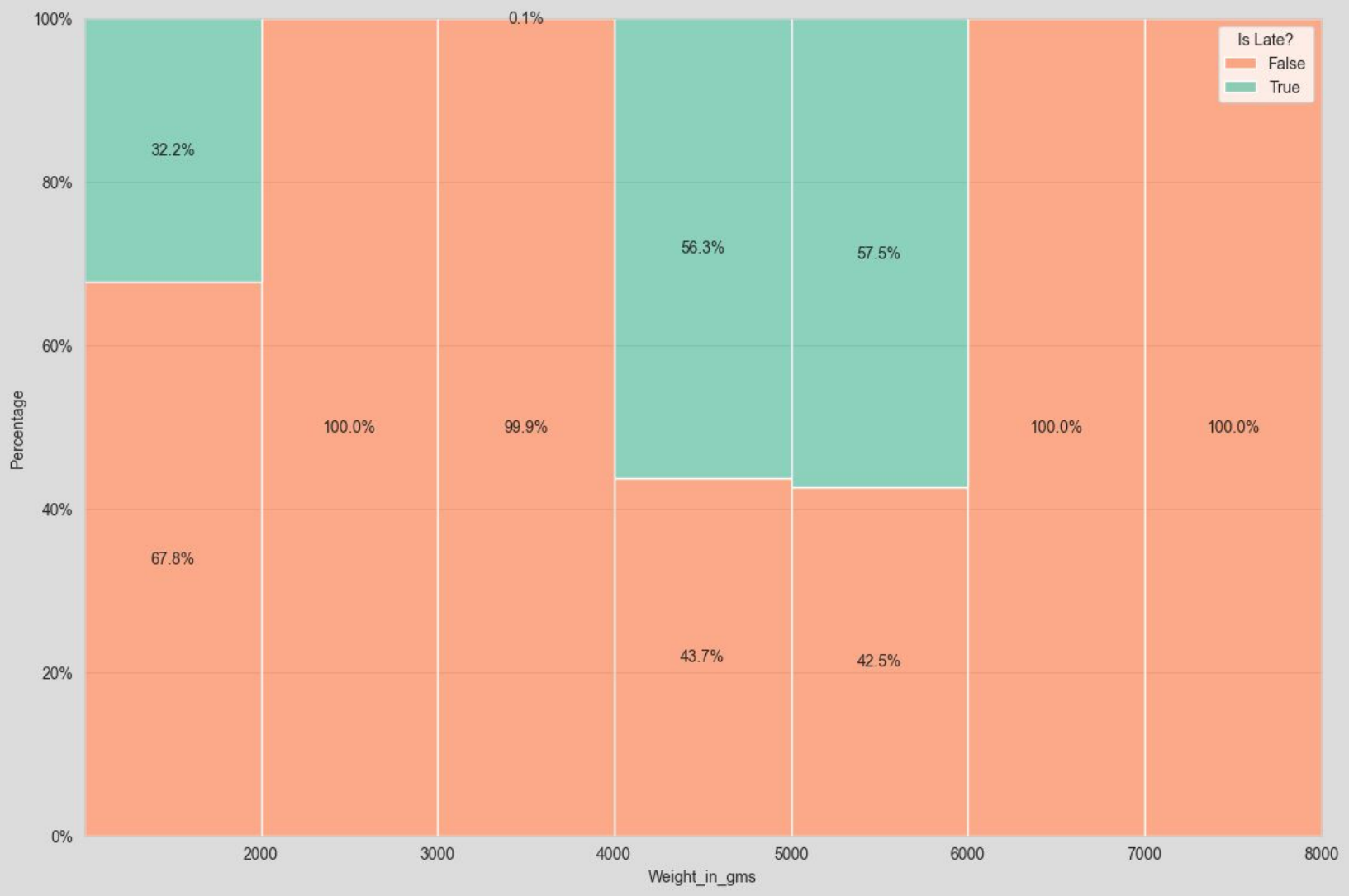
Diskon diatas 10% semuanya mengalami keterlambatan, hal ini kami asumsikan dikarenakan diskon produk yang diberikan tidak hanya memotong harga produk namun memotong shipment cost.

Rekomendasi

- Tidak memberikan diskon apabila akan memotong shipment costnya juga.
- Membatasi pemberian diskon hanya sebesar maximal 10%.

*disclaimer: semua produk memiliki diskon >= 1%

Stage 1 - Weight Problem



Insight

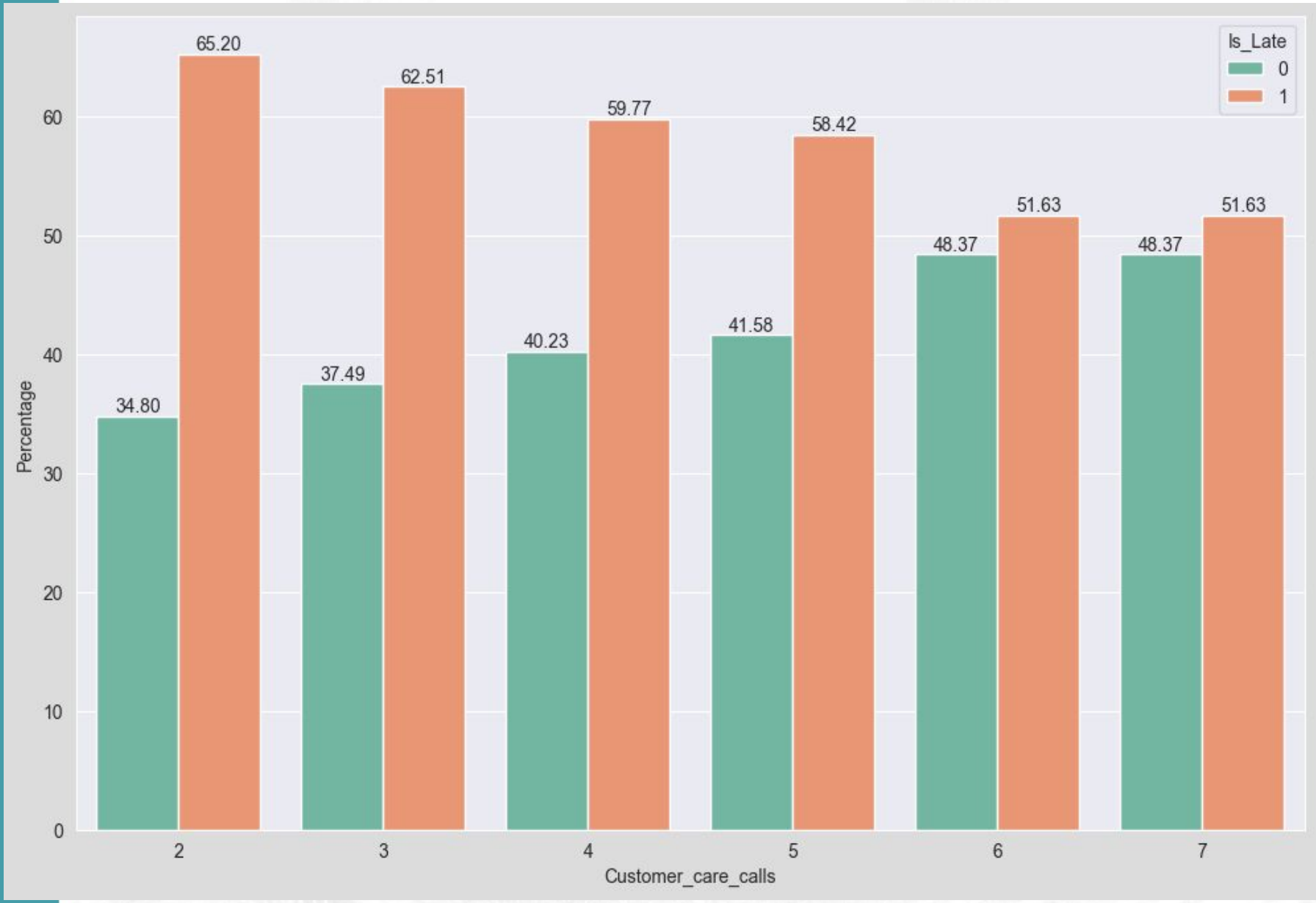
Berat 2-4 kg dan >6 kg semuanya mengalami keterlambatan hal ini mungkin dikarenakan Berat 2-4 kg merupakan berat yang nanggung (memiliki berat yang tidak ringan maupun berat namun memiliki shipment cost yang sama dengan 1000-2000). Sementara, Berat diatas 6 kg terlambat dikarenakan jumlah barang yang dapat diantar dalam satu kali pengiriman terbatas.

Rekomendasi

- Mengevaluasi kembali kategori shipment cost yang diberikan.

*disclaimer: semua produk memiliki berat >= 1 kg

Stage 1 - Calls Problem



Insight

Semakin sering customer menelpon, semakin rendah persentase keterlambatan. Hal ini diasumsikan karena hanya customer yang sering melakukan panggilan yang difollow up barangnya.

Rekomendasi

- Memperbaiki sistem antrian, jangan hanya memprioritaskan customer yang sering melakukan panggilan saja.

*disclaimer: customer sudah melakukan panggilan paling tidak 2x.

Stage 2 - Preprocessing

Pada stage 2 kami melakukan preprocessing sebagai berikut:

1. Handling data duplicate dan missing value
2. Handling outlier
3. Fitur Transformation
4. Fitur Encoding
5. Fitur Selection
6. Handling Imbalanced Data

Stage 2 - Handling data duplicate dan missing value

```
df.isna().sum()
```

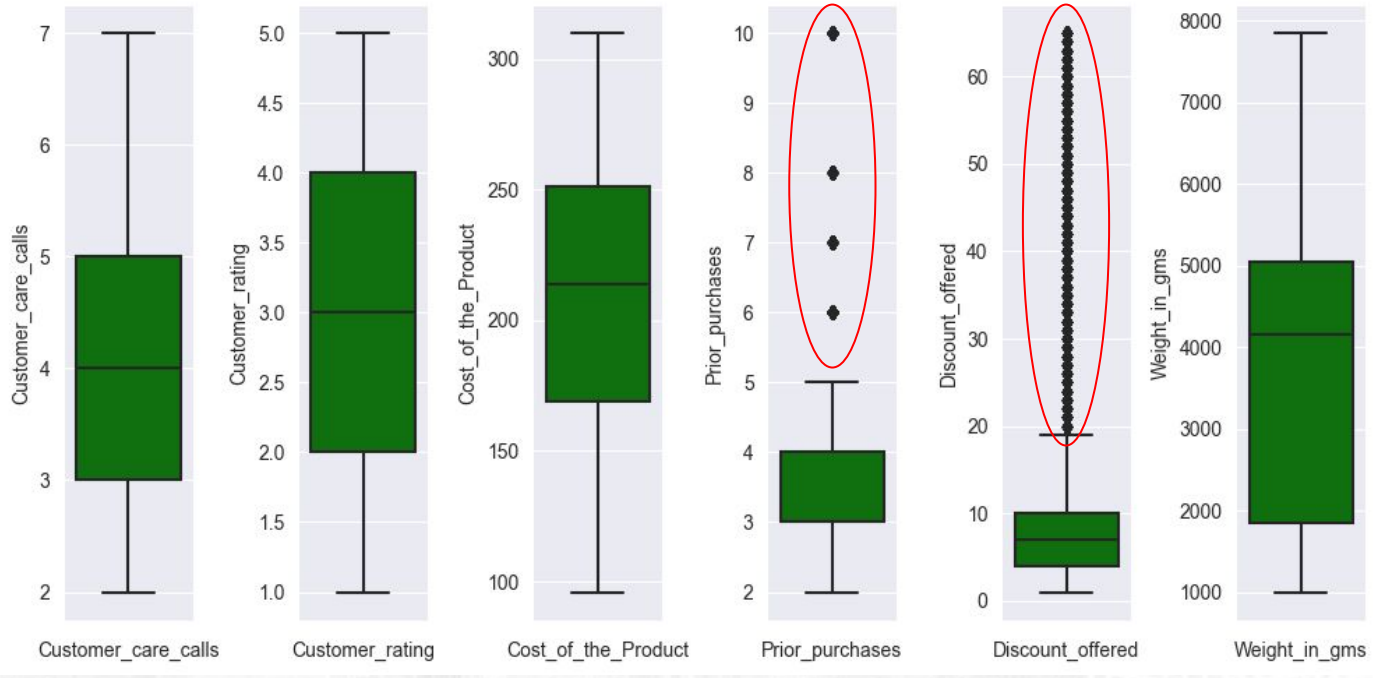
```
ID                      0
Warehouse_block         0
Mode_of_Shipment        0
Customer_care_calls     0
Customer_rating         0
Cost_of_the_Product     0
Prior_purchases         0
Product_importance      0
Gender                  0
Discount_offered        0
Weight_in_gms           0
Is_Late                 0
dtype: int64
```

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

```
0
```

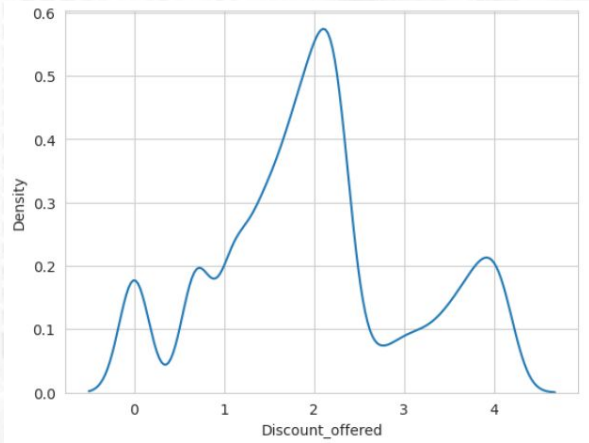
Tidak terdapat data duplikat dan missing value sehingga tidak perlu dilakukan action lanjutan.

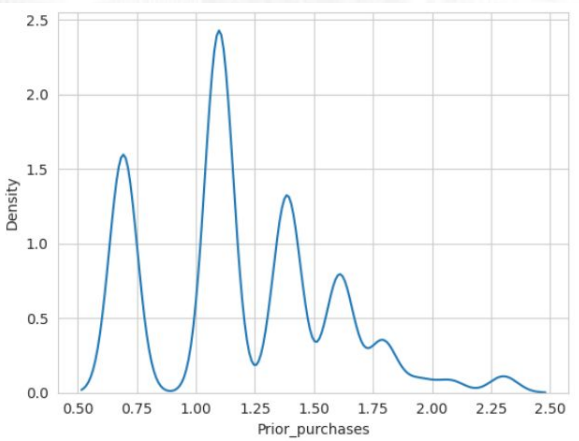
Stage 2 - Handling outlier



Terlihat adanya outlier yang terdapat pada kolom **Prior_purchases** dan **Discount_offered** sehingga perlu dihandling.

```
df['Prior_purchases'] = np.log(df['Prior_purchases'])
df['Discount_offered'] = np.log(df['Discount_offered'])
```





Jumlah data yang sedikit (10.999 data) menjadi pertimbangan dalam menggunakan *transformasi log* untuk handling outlier.
Jika menggunakan metode *IQR* dan *zscore* memungkinkan banyaknya data yang hilang

Stage 2 - Fitur Transformation

```
[ ]  #StandarScaler
     scaler = StandardScaler()
     scaler.fit(X_train[['Cost_of_the_Product']])
     X_train['Cost_of_the_Product'] = scaler.transform(X_train[['Cost_of_the_Product']])
     X_test['Cost_of_the_Product'] = scaler.transform(X_test[['Cost_of_the_Product']])

     #MinMaxScaler
     scaler = MinMaxScaler()
     scaler.fit(X_train[['Weight_in_gms']])
     X_train['Weight_in_gms'] = scaler.transform(X_train[['Weight_in_gms']])
     X_test['Weight_in_gms'] = scaler.transform(X_test[['Weight_in_gms']])
```

Scaler hanya dilakukan kepada kolom numerik yang memiliki range nilai jauh berbeda dengan kolom lainnya yaitu Cost_of_the_product dan Weight_in_gms. StandardScaler digunakan untuk kolom Cost_of_the_product karena datanya terdistribusi secara normal, sedangkan MinMaxScaler digunakan untuk kolom Weight_in_gms karena datanya tidak terlalu terdistribusi normal.

Stage 2 - Fitur Encoding

Terdapat 2 metode encoding yang dilakukan:

1. Label Encoding untuk kolom produk_importance (karena memiliki tingkatan high, medium, low) dan Gender (karena hanya memiliki 2 unik value)
2. One Hot Encoding untuk kolom Warehouse_block dan Mode_of_Shipment (karena tidak merepresentasikan tingkatan dan jumlah unik value lebih dari 2)

Warehouse_block_A	Warehouse_block_B	Warehouse_block_C	Warehouse_block_D	Warehouse_block_F	Mode_of_Shipment_Flight	Mode_of_Shipment_Road	Mode_of_Shipment_Ship
1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0
0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0
0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0

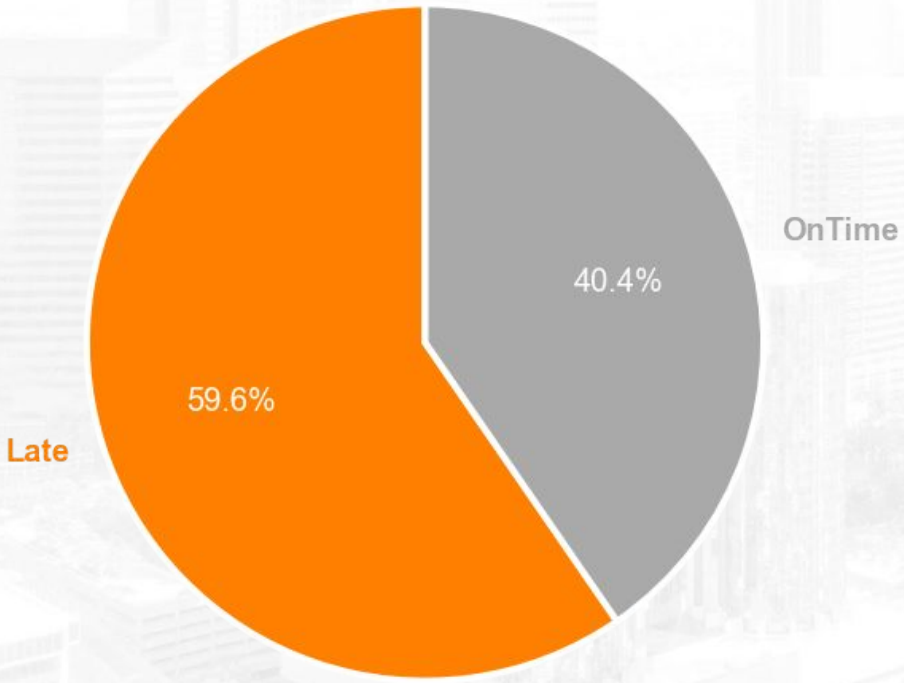
Product_importance	Gender
0	1
1	1
0	0
0	0
0	1

Stage 2 - Fitur Selection

Berdasarkan hasil Mutual Info Classification, Correlation, dan eksperimen feature importance kami mendapati bahwa kolom Warehouse_block memiliki nilai yang sangat rendah yang mana justru akan menurunkan performansi model yang dibangun, oleh karena itu kolom Warehouse_block akan didrop, sehingga fitur yang digunakan untuk model yaitu:

1. Discount_offered
2. Weight_in_gms
3. Customer_care_calls
4. Prior_purchases
5. Cost_of_the_Product
6. Product_Importance
7. Customer_rating
8. Gender
9. Mode_of_Shipment_Flight
10. Mode_of_Shipment_Ship
11. Mode_of_Shipment_Road

Stage 2 - Handling Imbalance Data



Degree of imbalance	Proportion of Minority Class
Mild	20-40% of the data set
Moderate	1-20% of the data set
Extreme	<1% of the data set

Berdasarkan laman **Google** (tabel di atas), *minority class* yang memiliki besar 40% termasuk ke dalam kategori **Mild** sehingga tidak diperlukan *handling imbalanced data*.

https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data

Stage 3 - Modelling Experiments

Tipe Machine Learning yang digunakan adalah **Classification (Supervised Learning)** untuk memprediksi target keterlambatan barang (Is_late). Tim kami menggunakan 6 model dan evaluasinya berikut :

Tipe Model

- Logistic Regression
- K-Nearest Neighbor
- Decision Tree
- Random Forest
- AdaBooost
- XGBoost

6 Model ini akan dievaluasi dengan evaluation metrics yang bertujuan untuk memastikan hasil prediksi model sudah mirip dengan data aktual atau tidak. Metrics yang digunakan adalah :

1. **Precision** (TP/TP+FP)
2. **Recall (TP/TP+FN)**
3. **Accuracy** (TP+TN/Total)
4. **F1 Score** (2 * Precision * Recall / Precision + Recall)
5. **Area under ROC curve (AUC)** TPR = (TP/TP+ FN) dan FPR = FP/(FP+FN)

Fokus tim kami adalah evaluasi **Recall** yang bertujuan untuk menurunkan *false negative* agar tidak terjadi kesalahan prediksi (barang aktual terlambat tapi diprediksi tidak terlambat) yang dapat mengakibatkan peningkatan ***late rate*** dan penurunan ***customer satisfaction.***

	Predicted True (1)	Predicted False (0)
Actual True (1)	True Positive (TP)	False Negative (FN)
Actual False (0)	False Positive (FP)	True Negative (TN)

Stage 3 - Hasil Modelling

Logistic Regression

```
Precision (train) : 0.6281254316894599
Precision (test)  : 0.6282404853833425
Recall (train)    : 0.8670861937452327
Recall (test)     : 0.8635329795299469
Accuracy (train)  : 0.6148425957495169
Accuracy (test)   : 0.6118181818181818
F1-Score (train)  : 0.7285107746535289
F1-Score (test)   : 0.7273307790549169
Auroc (train)     : 0.5549214372382985
Auroc (test)      : 0.549246625973827
```

K-Nearest Neighbor

```
Precision (train) : 0.8184870630061228
Precision (test)  : 0.7070467141726049
Recall (train)    : 0.7902364607170099
Recall (test)     : 0.6770280515542078
Accuracy (train)  : 0.7705421070576202
Accuracy (test)   : 0.6381818181818182
F1-Score (train)  : 0.8041137091297177
F1-Score (test)   : 0.691711851278079
Auroc (train)     : 0.7658636593317821
Auroc (test)      : 0.6285253765149018
```

Decision Tree

```
Precision (train) : 1.0
Precision (test)  : 0.7124528301886792
Recall (train)    : 1.0
Recall (test)     : 0.7156937073540561
Accuracy (train)  : 1.0
Accuracy (test)   : 0.6563636363636364
F1-Score (train)  : 1.0
F1-Score (test)   : 0.7140695915279879
Auroc (train)     : 1.0
Auroc (test)      : 0.6416152986259498
```

Random Forest

```
Precision (train) : 0.999809342230696
Precision (test)  : 0.6577639751552795
Recall (train)    : 1.0
Recall (test)     : 0.8028809704321456
Accuracy (train)  : 0.9998863507216729
Accuracy (test)   : 0.6313636363636363
F1-Score (train)  : 0.9999046620268853
F1-Score (test)   : 0.7231136906794128
Auroc (train)     : 0.99985935302391
Auroc (test)      : 0.5887276588823611
```

AdaBooost

```
Precision (train) : 0.5959768155472213
Precision (test)  : 0.5995454545454545
Recall (train)    : 1.0
Recall (test)     : 1.0
Accuracy (train)  : 0.5959768155472213
Accuracy (test)   : 0.5995454545454545
F1-Score (train)  : 0.7468489638966034
F1-Score (test)   : 0.7496447854504119
Auroc (train)     : 0.5
Auroc (test)      : 0.5
```

XGBoost

```
Precision (train) : 0.8605401732631222
Precision (test)  : 0.6826987307949232
Recall (train)    : 0.9660564454614798
Recall (test)     : 0.7748294162244125
Accuracy (train)  : 0.8864643709512444
Accuracy (test)   : 0.649090909090909
F1-Score (train)  : 0.910250651334112
F1-Score (test)   : 0.7258522727272727
Auroc (train)     : 0.8675570553608384
Auroc (test)      : 0.6178346854107306
```

Berdasarkan hasil evaluasi keenam model di slide sebelumnya, ***Logistic Regression*** dipilih sebagai model terbaik dengan nilai recall yang optimal dan wajar yaitu **86%** dan model tidak menunjukkan adanya overfit dengan nilai recall yang tidak berbeda jauh antara data train dan test. Setelah itu, model logistic regression dilakukan optimalisasi dengan *Hyperparameter tuning* (Grid Search CV) yang bertujuan untuk mencari kombinasi parameter terbaik secara menyeluruh. Hasilnya adalah sebagai berikut:

```
penalty = ['l1', 'l2', 'elasticnet', None]
C = [x for x in np.linspace(0, 2, 50)]
max_iter = [int(x) for x in np.linspace(50, 500, 50)]
hyperparameters_logistic = dict(penalty=penalty,
                                C=C,
                                max_iter=max_iter)
(GridSearchCV(LogisticRegression(random_state=3), hyperparameters_logistic, scoring='recall', cv=5), X_train, X_test, y_train, y_test)
```

Parameter optimal (*Best Params*):

C = 0.041
max_iter = 50
penalty = 12

Hasil evaluasi:

```
Precision (train) : 0.623411732900784
Precision (test)  : 0.619914346895075
Recall (train)    : 0.8794813119755912
Recall (test)     : 0.8779378316906747
Accuracy (train)  : 0.6115467666780315
Accuracy (test)   : 0.6040909090909091
F1-Score (train)  : 0.7296313874386964
F1-Score (test)   : 0.7267022278004392
Auroc (train)     : 0.5478981806010164
Auroc (test)      : 0.5360177240178685
```

Dengan melakukan *hyperparameter tuning* terjadi sedikit peningkatan nilai recall dari **86%** ke **88%** untuk data test.

STAGE3 - Business Simulation

Hasil Model Prediksi

- Customer late yang terprediksi late (TP) = 1158
- Customer late yang terprediksi on time (FN) =161
- Customer on time terprediksi on time (TN) = 171
- Customer on time terprediksi late (FP) = 710

Aktual on time: 881
Aktual late 1319

Potential Revenue loss

Total Sales = jumlah Cost_of_the_Product = $458964
Total Diskon = jumlah harga diskon = $59048
Total Revenue = Total Sales - Total Diskon = $399916
Revenue per product = Total Revenue/Jumlah product
= $181.78
Potential Revenue Loss = Revenue per product * aktual late
= $239767.87

Hasil Model Prediksi

Berdasarkan www.freightos.com harga metode shipment adalah sebagai berikut:
Road = 2$/kg
Ship = 4$/kg
Flight = 8$/kg
Total Shipment Cost = $39536 atau $18 per product

Potential Revenue loss

Agar product dapat dikirim secara on time maka biaya yang perlu dikeluarkan adalah 2x biaya normal. Perusahaan memberikan budget sebesar $50000 untuk mengurangi late rate yang terjadi.

STAGE3 - Business Simulation (2)

Jumlah Produk yang dapat diberikan tambahan

Harga per produk agar on time = shipment cost * 2

= 18*2 = $36

Produk yang dapat diberikan tambahan biaya

= Budget Total / 36

= 50000/36 = 1389 Barang

Jumlah Barang yang terlambat

Jumlah produk terlambat sebelumnya = 1319 (59.9%)

Jumlah produk terlambat setelah diberikan treatment

= Total product - (1389 + TN)

= 2200 - (1389 + 171)

= 640 (29%)

Penurunan late rate= 1319 - 640 = 679 (51.4%)

Potential Revenue Loss After Treatment

Potential Revenue Loss After Treatment = Revenue per product * jumlah barang late

= 181.78 * 640

= $116339.2

Penurunan Potential Revenue Loss = Potential Revenue Loss Before Treatment - Potential Revenue Loss After Treatment

= $239767.87 - $116339.2

= $123428.64(51.4%)

STAGE3 - Business Recommendation

Berdasarkan dari insight dan problem yang ditemukan sebelumnya, kami merumuskan beberapa business recommendation diantaranya sebagai berikut:

1. Late Notification dan Tracking Location

Untuk barang yang tidak diberikan treatment dan terprediksi mengalami keterlambatan dapat diberikan notifikasi bahwa barang akan terlambat dan tracking location untuk tetap menjaga customer satisfaction.

2. Discount Optimization

Mengoptimisasi diskon yang diberikan sehingga nantinya selisih uang antara diskon yang di optimisasi dengan yang tidak, dapat digunakan untuk biaya tambahan shipment.

3. Increase Handling Time and ManPower

- Penambahan man power.
- Memperbaiki sistem antrian agar memberikan prioritas yang adil.
- Untuk package dengan berat <= 2.000 dan waktu order diterima < pukul 11.00, maka waktu handling < 1 hari.
- Penetapan waktu maksimal handling package adalah 1 hari setelah order diterima.
- Same-day delivery 75 packages/hari dapat menurunkan late rate hingga 34%.

Pembagian Tugas

Stage 0:

- Semua anggota berdiskusi dan mengerjakan secara bersama-sama.

Stage 1:

- Semua anggota mencari insight masing-masing yang kemudian digabungkan untuk mencari berbagai insight dan perspektif

Stage 2:

- Eksplorasi berbagai attributes, Mengecek apakah ada data bermasalah : Nabil
- Handling missing value, Handling duplicated data, Handling outlier data : Riel
- Feature Transformation (Numeric), Feature Encoding (Categoric) : Febi
- Feature extraction, Feature selection : Vicky
- Handling imbalanced data : Qistina
- Feature Tambahan : Kevin

Pembagian Tugas

Stage 3:

- Semua anggota mengerjakan masing-masing kemudian didiskusikan dan digabung bersama sama

Stage 4:

PPT, presentasi, dan laporan dibagi dengan pembagian sebagai berikut:

- Perkenalan Hexa, Latar Belakang Masalah : Qistina
- Insight : Nabil
- Preprocessing: Kevin
- Metrics evaluasi yang digunakan, Modeling, Hyperparameter tuning : Riel
- Feature Importance, Rekomendasi Bisnis, Simulasi Rekomendasi (2 Orang) : Vicky & Febiya