Rakamin Academy

Team Cobra (Kelompok 4)

Dokumen Laporan Final Project

Rakamin Academy

STAGE 0
PREPARATION

Latar Belakang Masalah

Sebuah perusahaan e-commerce skala internasional seringkali mengalami masalah **keterlambatan** dalam proses pengiriman barang. Hal ini dapat berdampak kepada tingkat kepuasan pelanggan (*customer satisfaction*) dan memungkinkan terjadinya penurunan total penjualan. Perusahaan membutuhkan sebuah **model *machine learning*** yang dapat memprediksi apakah suatu pengiriman akan terlambat atau tidak sebagai upaya untuk meningkatkan rasio pengiriman tepat waktu.

Peran Tim Kami

Tim kami berperan sebagai ***Data Scientists*** dalam perusahaan e-Commerce Internasional yang menjual produk elektronik, yang ditugaskan untuk melakukan analisis data dan membangun model machine learning yang dapat membantu perusahaan e-commerce dalam memprediksi keterlambatan pengiriman dan meningkatkan customer satisfaction untuk meningkatkan penjualan

Tujuan

- Mengurangi rasio keterlambatan pengiriman barang dari sekitar 40% menjadi 20% pada akhir tahun

- Meningkatkan total sales sebesar 10-20% dibanding tahun sebelumnya dengan meningkatnya jumlah order customer, seiring dengan meningkatnya kepuasan customer terhadap pengiriman barang

Sasaran

- Mencari fitur-fitur yang memiliki korelasi/hubungan dengan keterlambatan pengiriman barang
- Membuat model *machine learning* untuk memprediksi apakah suatu pengiriman akan terlambat atau tidak
- Memberikan rekomendasi tindakan terhadap pengiriman yang diduga memiliki potensi keterlambatan yang tinggi

Business Metrics

Main metric: Delivery on time ratio

Metric ini menghitung jumlah pengiriman barang yang tiba tepat waktu dibagi dengan total pengiriman barang dalam periode waktu tertentu, sehingga dapat memberikan gambaran tentang seberapa efektif perusahaan dalam memenuhi kewajiban pengiriman barang tepat waktu.

Supporting metric: Customer satisfaction (rating/call/complaints)

Metric ini dapat dihitung dengan memperhatikan beberapa faktor, seperti rating yang diberikan oleh pelanggan, jumlah panggilan layanan pelanggan, dan jumlah keluhan yang diterima oleh perusahaan. Dengan metric ini perusahaan dapat mengetahui tingkat kepuasan pelanggan dan mengambil tindakan yang sesuai untuk meningkatkan pengalaman pelanggan.

STAGE 1
Exploratory Data Analysis

Descriptive Statistics

Column	Description	Non-Null Count	Data Type	Categories of Data
ID	ID Number of Customers.	10999 non-null	int64	Discrete Data
Warehouse_block	The Company have big Warehouse which is divided in to block such as A,B,C,D,E.	10999 non-null	object	Nominal Data
Mode_of_Shipment	The Company Ships the products in multiple way such as Ship, Flight and Road.	10999 non-null	object	Nominal Data
Customer_care_calls	The number of calls made from enquiry for enquiry of the shipment.	10999 non-null	int64	Discrete Data
Customer_rating	The company has rated from every customer. 1 is the lowest (Worst), 5 is the highest (Best)	10999 non-null	int64	Ordinal Data
Cost_of_the_Product	Cost of the Product in US Dollars.	10999 non-null	int64	Continuous Data
Prior_purchases	The Number of Prior Purchase.	10999 non-null	int64	Discrete Data
Product_importance	The company has categorized the product in the various parameter such as low, medium, high.	10999 non-null	object	Nominal Data
Gender	Male and Female.	10999 non-null	object	Nominal Data
Discount_offered	Discount offered on that specific product.	10999 non-null	int64	Discrete Data
Weight_in_gms	Weight of product in grams	10999 non-null	int64	Continuous Data
Reached.on.Time_Y.N	Boolean of whether shipment reached in time or not (0 = No, 1 = Yes)	10999 non-null	int64	Nominal Data

Tidak ada null values dan **tidak ada duplicate ID** (setiap baris adalah unique customer)

Descriptive Statistics - Numerical Data

	Customer_care_calls	Customer_rating	Cost_of_the_Product	Prior_purchases	Discount_offered	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	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

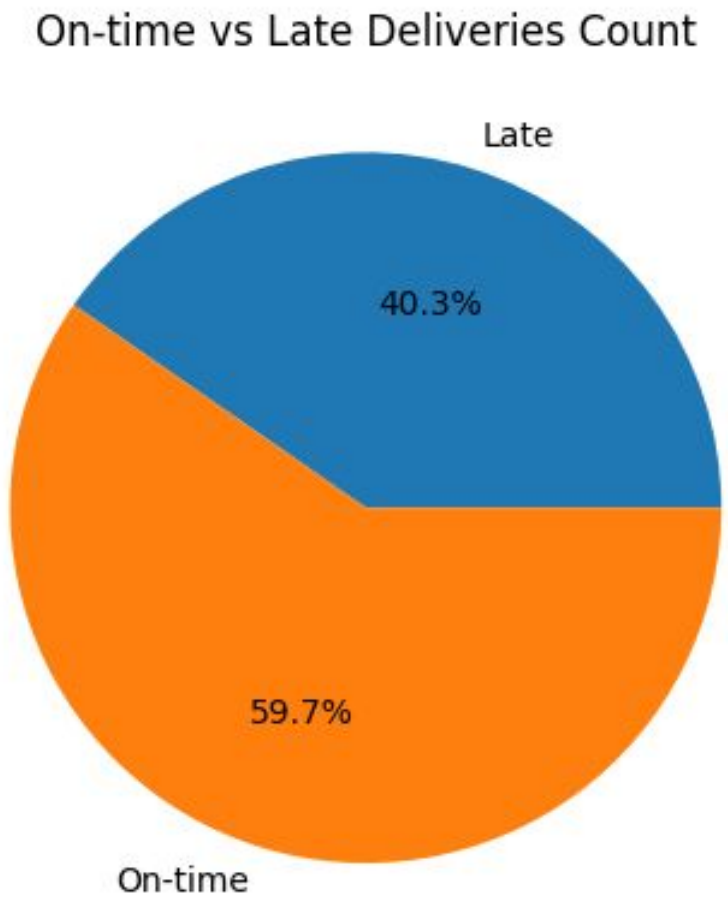
Terdapat sebuah kejanggalan pada feature ***Discount_offered***, dimana nilai maksimalnya sebesar **65** sedangkan selisih jaraknya dengan **Q3 (10)** atau **mean (13)** sangat tinggi, sehingga diduga terdapat beberapa ***outliers***

Descriptive Statistics - Categorical Data

	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

Tidak terdapat kejanggalan pada *Categorical Data* karena setiap kolom memiliki jumlah data yang sama dengan jumlah total baris data dan tidak terdapat nilai yang tidak diharapkan seperti huruf atau karakter yang dianggap sebagai kejanggalan.

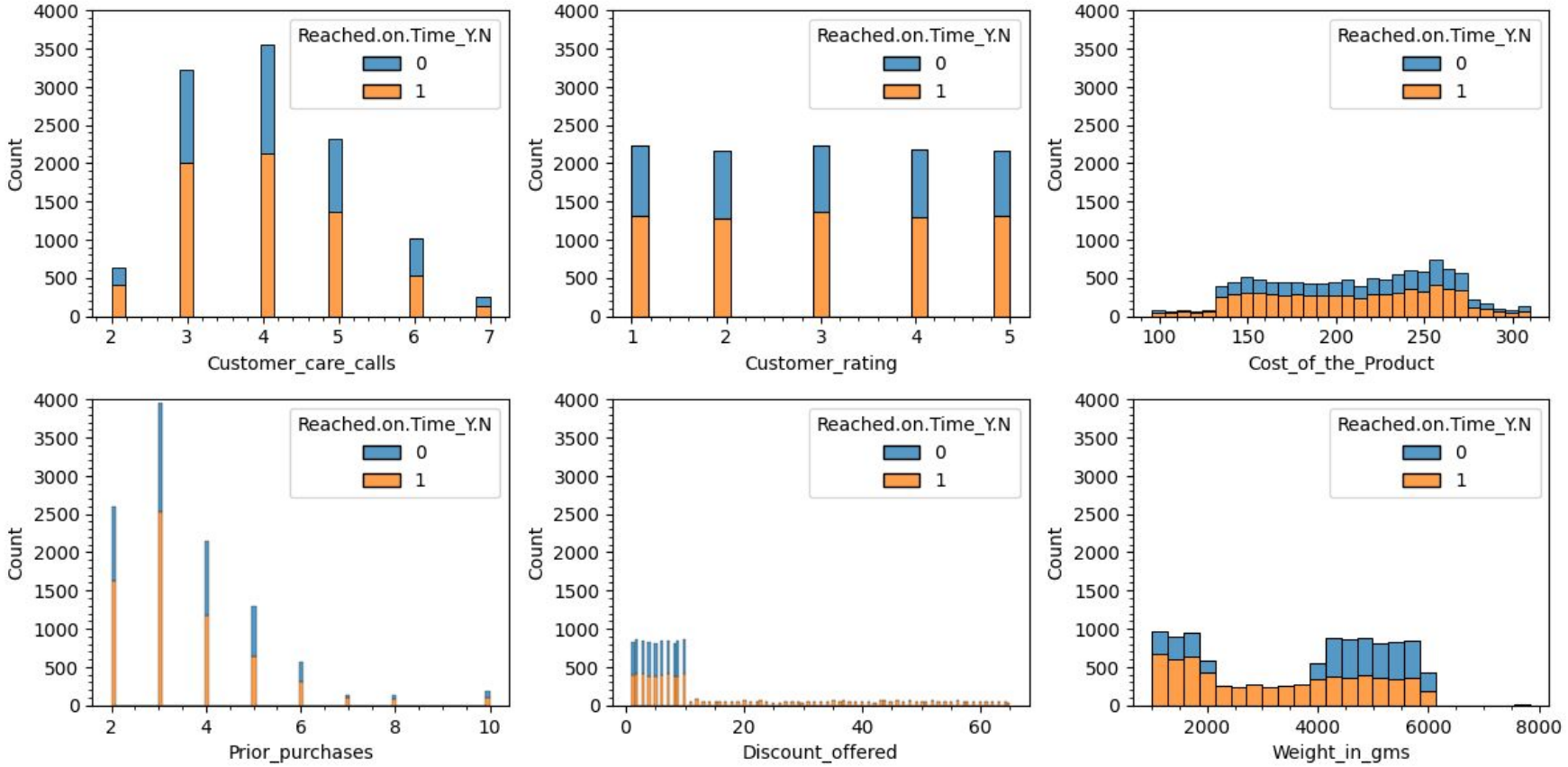
Univariate Analysis



Diketahui bahwa proporsi pengiriman yang tepat waktu (*On-time*) sebesar 59,7% dan proporsi pengiriman yang terlambat (*Late*) sebesar 40,3%. Dapat disimpulkan bahwa sebagian besar pengiriman dilakukan tepat waktu, namun masih ada sebagian kecil pengiriman yang terlambat. Hal ini dapat menjadi fokus perbaikan untuk meningkatkan kualitas layanan dan kepuasan pelanggan.

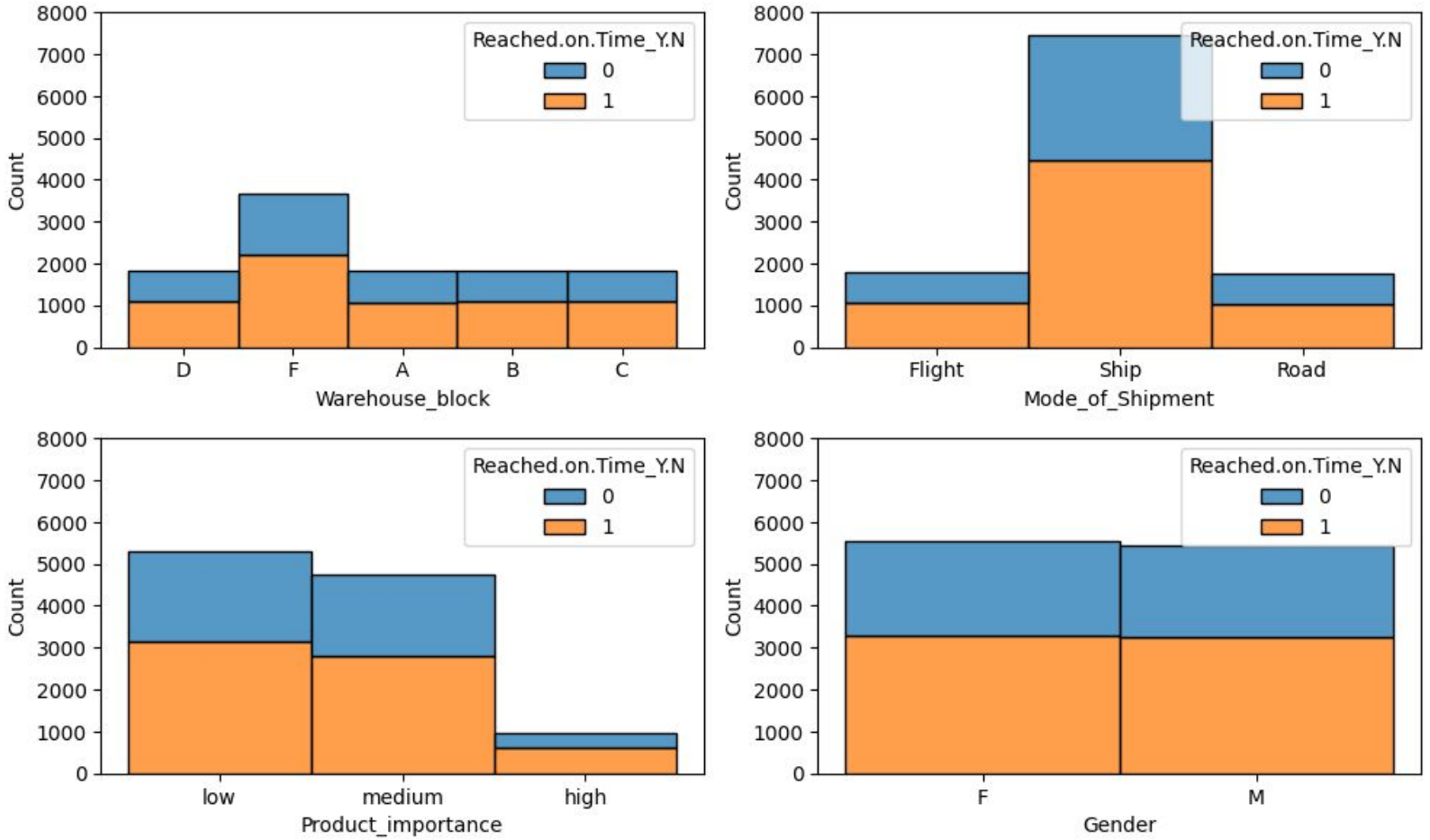
Karena data sudah memiliki jumlah sampel yang cukup banyak dan tidak terlalu berbeda antara kelas satu dengan yang lain (tidak terlalu signifikan *class imbalance*-nya), maka **tidak perlu dilakukan penyeimbangan lebih lanjut** pada tahap *pre-processing* data.

Univariate Analysis



- Pada feature *Prior_purchase* membentuk positive skew
- Pada feature *Weight_in_gms* juga terdapat beberapa outlier pada nilai diatas 7500 (tidak terlalu terlihat pada grafik)
- Pada feature *Discount_offered*, terdapat nilai yang mendominasi yaitu pada nilai 0 - 10
- Saat data pre-processing, perlu dilakukan scaling pada data numerik agar mempunyai range yang seragam.

Univariate Analysis



- Berdasarkan warehouse, barang paling banyak disimpan/dilayani oleh warehouse F, sedangkan warehouse lainnya kurang lebih menampung jumlah barang yang sama
- Mayoritas pengiriman dilakukan melalui jalur laut (Ship)
- Jumlah barang yang tingkat kepentingannya tinggi (high) relatif sedikit
- Jumlah customer pria hampir setara dengan jumlah customer wanita, dengan rasio keterlambatan yang juga serupa

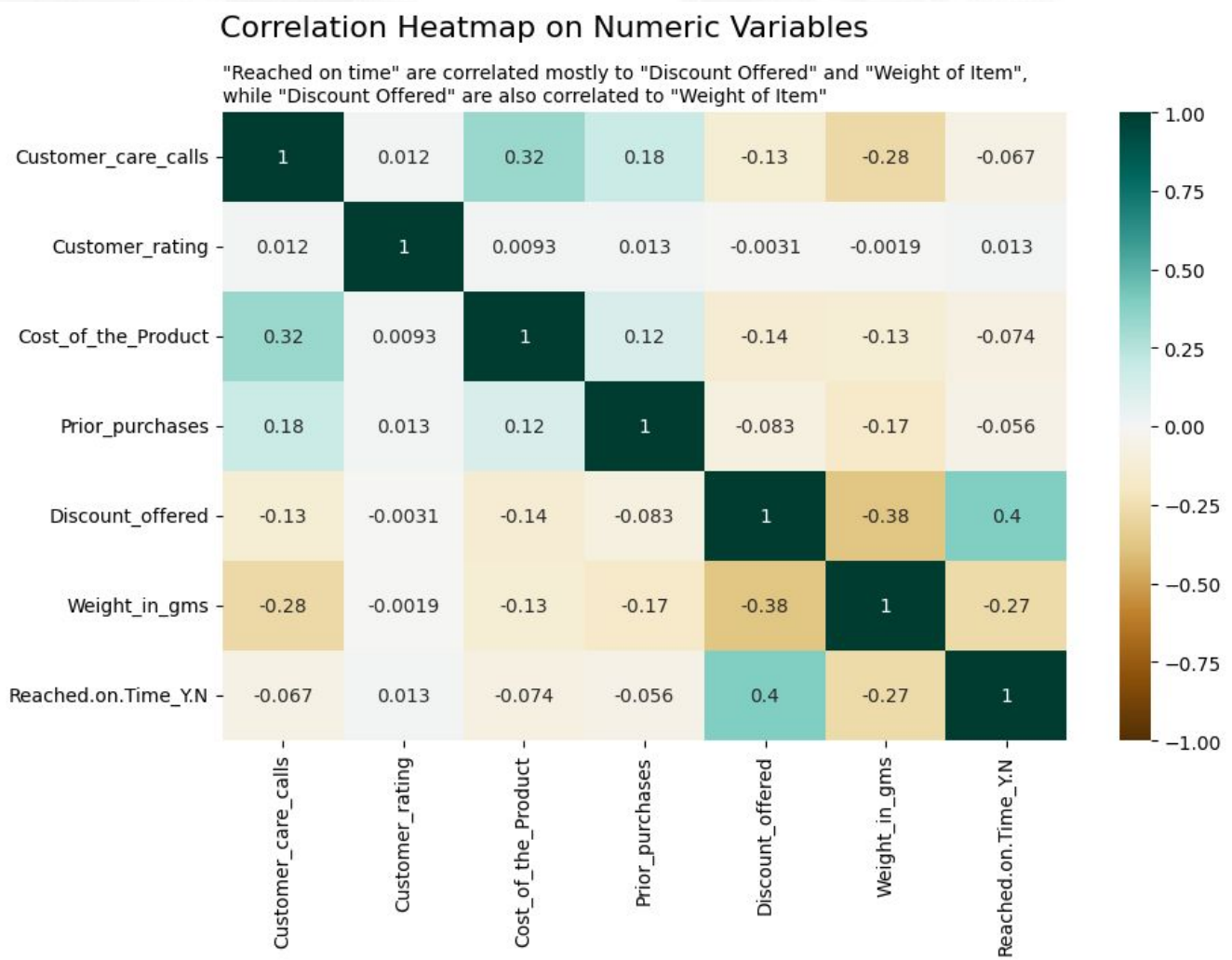
Univariate Analysis: Next Steps

- Beberapa hal yang harus ditindaklanjuti pada saat data *pre-processing*:
  - Untuk kolom Gender, dapat dilakukan mapping agar nilai "M" dan "F" menjadi 0 dan 1.
  - Untuk kolom Product_importance, dapat dilakukan label encoding karena kedua kolom tersebut memiliki nilai-nilai yang berurutan dan memiliki hubungan ordinal antara nilai-nilainya.
  - Untuk kolom Warehouse_block dan Mode_of_Shipment, dapat dilakukan one-hot encoding karena kolom tersebut tidak memiliki hubungan ordinal antara nilai-nilainya dan setiap nilai kategorikal dianggap sama pentingnya.

Multivariate Analysis - Correlation

Berdasarkan hasil heatmap yang dibuat korelasi antar feature beragam dengan range 1 sampai -1. Semakin mendekati 1 atau -1 maka korelasi semakin kuat, sedangkan semakin mendekati 0 maka korelasi semakin lemah. Beberapa nilai korelasi yang paling relevan adalah sebagai berikut:

- *Discount_offered* dengan *Reached.on.Time_Y.N* berkorelasi sedang positif, sedangkan *Weight_in_gms* dengan *Reached.on.Time_Y.N* berkorelasi lemah negatif
- *Discount_offered* dengan *Weight_in_gms* berkorelasi sedang negatif (diduga dapat mengakibatkan **multikolinearitas**)
- *Cost_of_the_Product* dengan *Customer_care_calls* berkorelasi sedang positif
- *Customer_rating* memiliki korelasi yang sangat kecil terhadap seluruh fitur lainnya, termasuk keterlambatan pengiriman



Multivariate Analysis - Correlation

- Berdasarkan hasil korelasi antara feature, terdapat beberapa hal yang perlu dilakukan, yaitu:
    - *Discount_offered* dan *Weight_in_gms* memiliki korelasi yang cukup signifikan, sehingga perlu dilakukan pengecekan terhadap adanya multikolinearitas antara kedua feature tersebut. Jika ditemukan adanya multikolinearitas, salah satu feature dapat dihapus atau digabungkan dengan feature lain.
    - Korelasi antara *Cost_of_the_Product* dengan *Customer_care_calls* perlu diperhatikan pada saat modelling. Jika terdapat multikolinearitas, feature yang memiliki korelasi lebih rendah dengan target (*Reached.on.Time_Y.N*) dapat dihapus atau digabungkan dengan feature lain.
    - Feature yang memiliki korelasi rendah dengan target (*Reached.on.Time_Y.N*) namun memiliki korelasi yang tinggi dengan feature lain juga perlu diperhatikan pada saat modelling. Pada beberapa kasus, feature tersebut mungkin dapat dihapus atau digabungkan dengan feature lain untuk menghindari multikolinearitas dan meningkatkan akurasi model.

Multivariate Analysis - Pair Plot

Terdapat segmentasi data yang secara visual cukup jelas terlihat pada beberapa pair plot. Secara visual, dapat diambil beberapa insight sebagai berikut:

- Pada *discount offered* terhadap *weight*, pada umumnya barang-barang yang beratnya di atas 4000 gram tidak diberikan diskon lebih besar dari 10% (kecuali untuk beberapa outlier).
- **Tidak ditemukan barang terlambat** pada barang yang diberikan **diskon lebih dari 10%**
- Barang dengan **berat di antara 2000-4000 gram** harganya ada di kisaran ~200 sampai ~300 dollar, dan **tidak ada yang terlambat pengirimannya**
- Terdapat beberapa data outlier jika dilihat berdasarkan berat barang (Weight_in_gms), yaitu barang-barang yang beratnya melebihi 6000 gram, namun untuk barang-barang tersebut **tidak ada satupun yang mengalami keterlambatan**

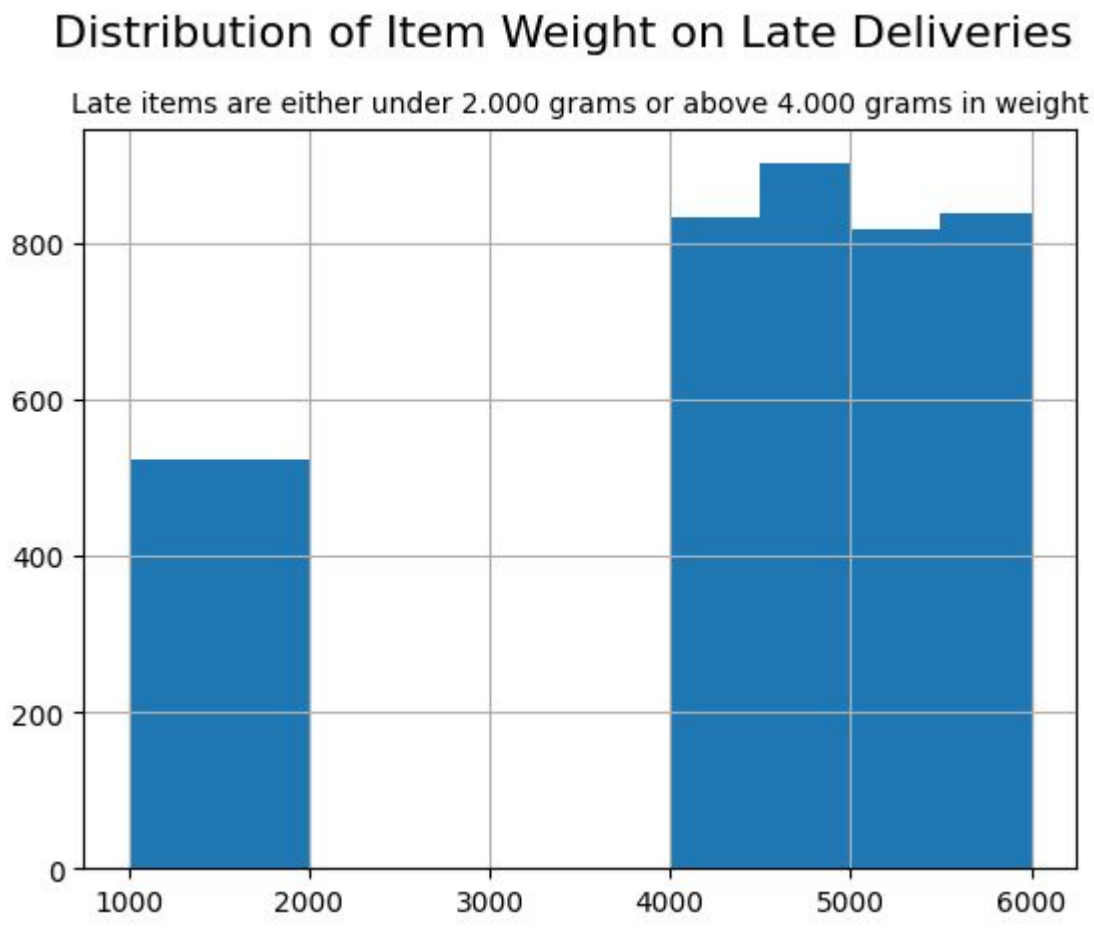
Multivariate Analysis - Pair Plot

- Berdasarkan hasil korelasi antara feature, terdapat beberapa hal yang perlu dilakukan, yaitu:
    - Beberapa data outlier pada berat barang (*Weight_in_gms*) yang perlu diobservasi lebih lanjut untuk memastikan apakah data tersebut valid atau tidak. Jika data tersebut valid, maka dapat dipertimbangkan untuk menggunakan teknik *pre-processing* seperti pengurangan dimensi (PCA) atau penanganan outlier untuk memperbaiki performa model.

Business Insights

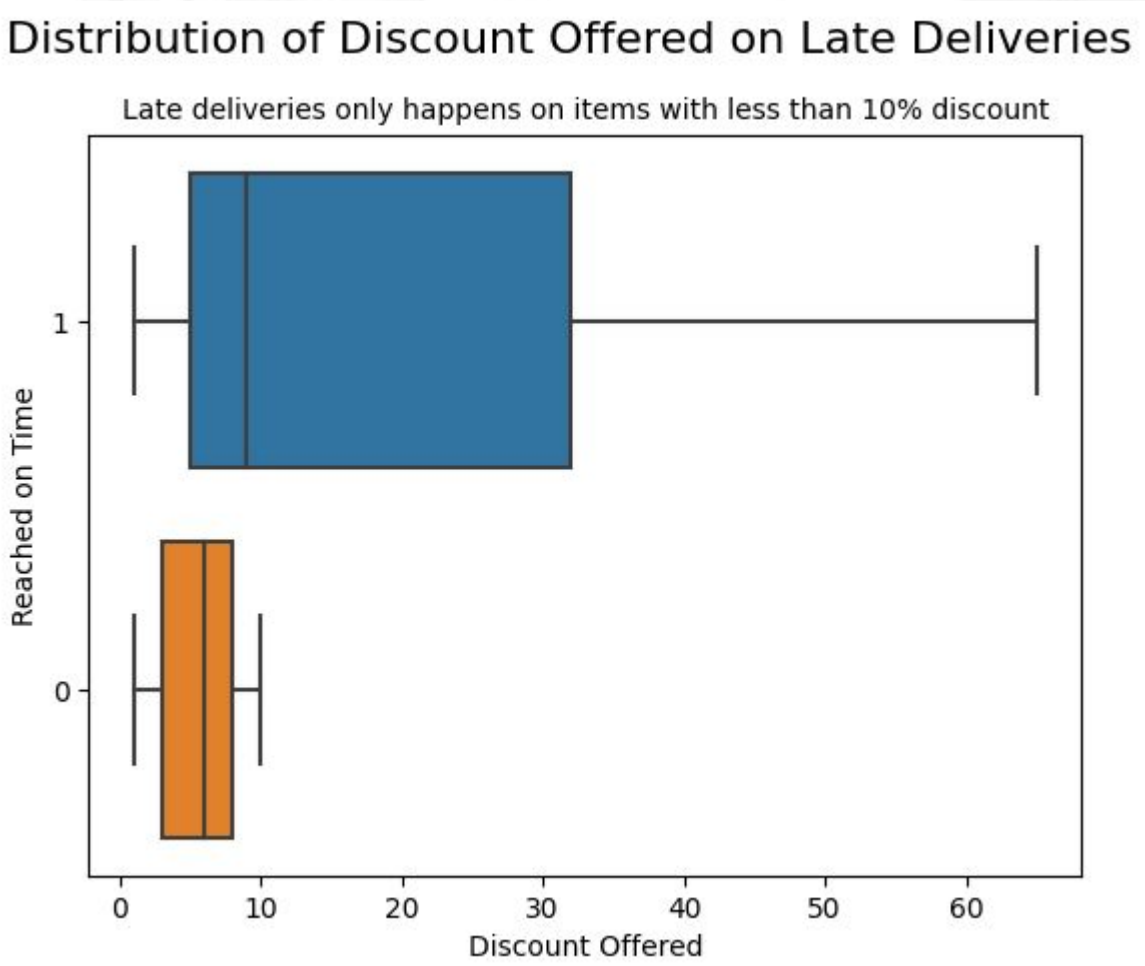
- Berdasarkan data yang tersedia, rasio keterlambatan pengiriman berada di 40.3% dari total pengiriman
- Beberapa fitur yang diduga memiliki korelasi dengan keterlambatan pengiriman adalah:
  - Berat barang (*Weight_in_gms)*
  - Diskon (*Discount_offered*)
- Terdapat beberapa data outlier jika dilihat berdasarkan berat barang (*Weight_in_gms*), yaitu barang-barang yang beratnya melebihi 6000 gram, namun untuk barang-barang tersebut tidak ada satupun yang mengalami keterlambatan (walaupun jumlah sampel outlier tersebut tidak banyak untuk dapat mengambil kesimpulan yang bermakna)
- Nilai *Customer rating* tidak memiliki korelasi dengan keterlambatan (*Reached_on_time*) bahkan juga dengan *feature-feature* lain.

Business Insights - Recommendations



- Keterlambatan hanya terjadi pada barang yang beratnya ada **di bawah 2000 gram** atau di antara **4000-6000 gram**. Diperlukan analisis lanjutan untuk mengetahui penyebab keterlambatan. Bisa jadi, perusahaan tidak memiliki partner shipment yang secara khusus mengantarkan paket produk untuk perusahaannya. Sehingga pengiriman barang bercampur dengan perusahan lain berdasarkan berat barang, *mode of shipment* ataupun faktor lain dan akhirnya terjadi keterlambatan.

Business Insights - Recommendations



- Rekomendasi : memberikan **pengawasan** / **perhatian** khusus pada **barang-barang** yang diberikan **discount <10%**. Perlu diperiksa apakah ada ketentuan perusahaan atau situasi yang berbeda pada barang-barang yang diberikan diskon dibawah 10% dibandingkan dengan jumlah lainnya, yang dapat mempengaruhi keterlambatan

Git

Link Repository Github: https://github.com/mezkymy/ecommerce-ds

STAGE 2
Data Preprocessing

Data Cleansing

Missing Values & Duplicate Data

Column	Non-Null Count
ID	10999 non-null
Warehouse_block	10999 non-null
Mode_of_Shipment	10999 non-null
Customer_care_calls	10999 non-null
Customer_rating	10999 non-null
Cost_of_the_Product	10999 non-null
Prior_purchases	10999 non-null
Product_importance	10999 non-null
Gender	10999 non-null
Discount_offered	10999 non-null
Weight_in_gms	10999 non-null
Reached.on.Time_Y.N	10999 non-null

Seperti yang telah ditunjukkan pada Stage 1 (Exploratory Data Analysis), melalui penggunaan metode ***info()*** dan juga pemeriksaan ID duplikat **tidak ditemukan data yang kosong** pada kolom manapun dan juga **tidak ditemukan duplikasi** pada kolom ID.

Dengan demikian, disimpulkan bahwa data **tidak perlu diolah lebih lanjut terkait data kosong dan duplikat**.

```
[ ]   # check if any ID is duplicated
      df[df['ID'].duplicated()]
```

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	Reached.on.Time_Y.N

No duplicate ID detected.

Data Cleansing

Train-Test Split

Sebelum melakukan transformasi data lebih lanjut, ada baiknya data dipisah terlebih dahulu supaya pemrosesan data dapat dilakukan hanya kepada data *train* saja terlebih dahulu untuk mencegah terjadinya *data leakage*. Setelah itu, transformasi data yang sama yang telah diterapkan pada data *train* dapat digunakan pada data *test*.

Berikut adalah langkah-langkah yang diambil pada train-test split:

1. Lakukan split antara target dengan fitur terlebih dahulu (x_data, y_data)
2. Split dengan rasio 80:20, menggunakan parameter random state = 25 (untuk memastikan split data sama tiap kali cell dijalankan) dan stratify = y_data (untuk memastikan pembagian dilakukan secara merata berdasarkan target)
3. Pastikan train & test sudah terbagi dengan tepat.

```
# split data into features & target
target = 'Reached.on.Time_Y.N'
features = df.loc[:, df.columns != target].columns
x_data = df[features]
y_data = df[target]
```

```
# split data menjadi 80% data train dan 20% data test

x_train, x_test, y_train, y_test = train_test_split(x_data, y_data, test_size=0.2, random_state=25, stratify=y_data)

# check amount of data on train and test
print('banyaknya data train =', x_train.shape[0])
print('banyaknya data test =', x_test.shape[0])

# check if split is balanced based on target value
print('mean value of y on train =', y_train.mean())
print('mean value of y on test =', y_test.mean())
```

```
banyaknya data train = 8799
banyaknya data test = 2200
mean value of y on train = 0.5966587112171837
mean value of y on test = 0.5968181818181818
```

Hasil akhir yang diperoleh adalah data yang terbagi menjadi

8799 data train dan **2200 data test**

Data Cleansing

Outliers

Pada tahapan selanjutnya, data train diperiksa untuk melihat apakah terdapat outlier yang perlu ditangani supaya pembentukan model dapat dilakukan lebih baik. Pemeriksaan outlier dilakukan hanya kepada data train.

Untuk memeriksa outlier, digunakan dua metode yaitu berdasarkan z-score dan berdasarkan interquartile range (IQR) yang dimodifikasi. Penentuan batas atas dan batas bawah menggunakan IQR dimodifikasi dari Q1 - (IQR * 1.5) dan Q3 + (IQR * 1.5) menjadi Q1 - (IQR * 2) dan Q3 + (IQR * 2) untuk memperkecil jumlah data yang dianggap sebagai outlier.

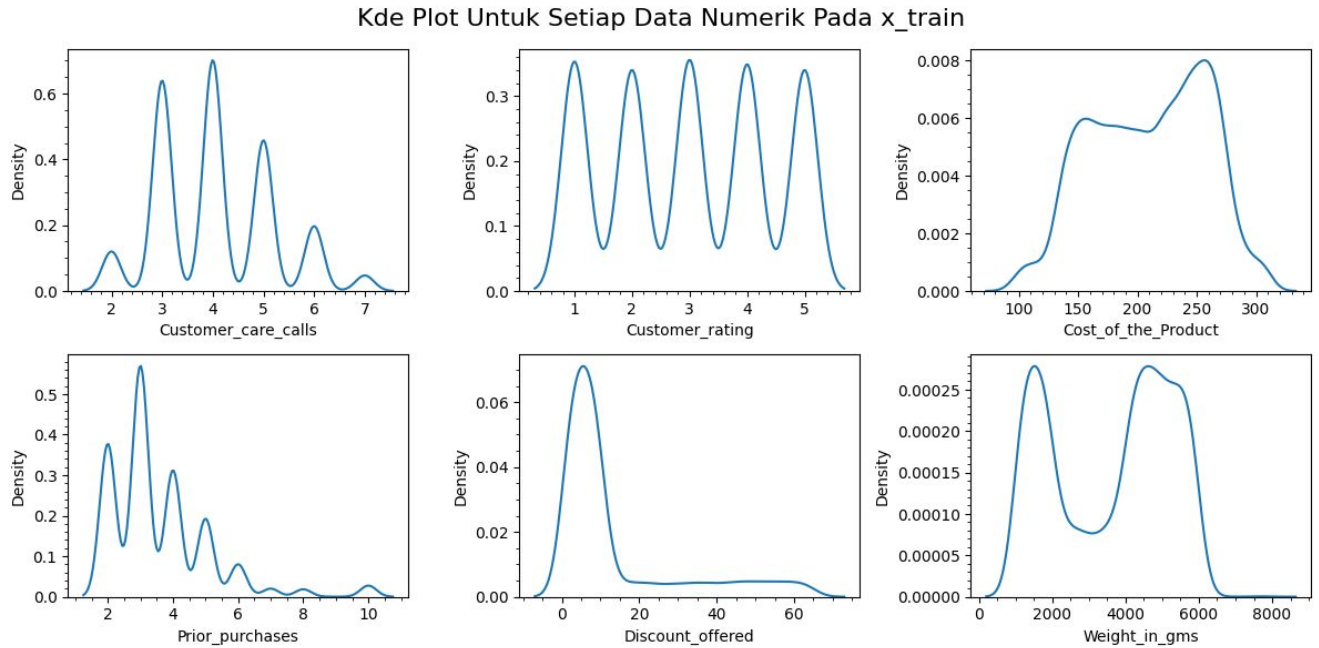
Berdasarkan nilai **z-score**, terdapat **151 data** yang memiliki nilai outlier pada *Prior_purchases* atau *Discount_offered*, sedangkan berdasarkan **IQR** yang dimodifikasi terdapat **362 data** outlier. Kedua metode mendeteksi outlier kurang dari 5% dari jumlah data train, sehingga dirasa aman untuk dihilangkan dengan metode manapun.

Data Cleansing

Feature Transformation

Berikut adalah langkah-langkah yang diambil pada tahapan Feature Transformation:

1. Memilih kolom-kolom numerik pada data train yang ingin ditinjau distribusinya dan menghitung *skewness* dari tiap data numerik.
2. Hasilnya kolom *Customer_care_calls*, *Customer_rating*, *Cost_of_the_Product*, dan *Weight_in_gms* dapat diasumsikan memiliki distribusi yang normal. Sehingga akan dilakukan transformasi untuk fitur lainnya yaitu *Prior_purchases* dan *Discount_offered*.
3. Melakukan transformasi dan scaling pada dua fitur, yaitu *Prior_purchases* dan *Discount_offered*, dengan menggunakan Normalizer dan Log Transformation.

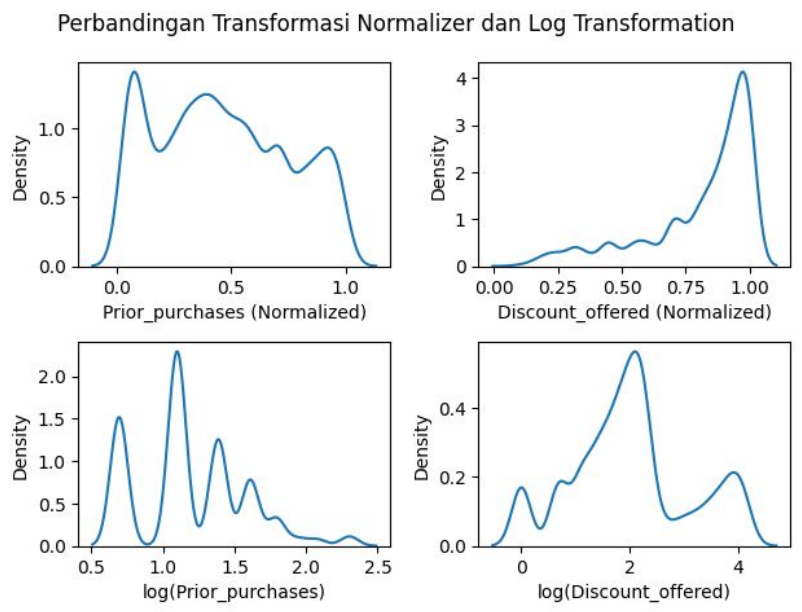


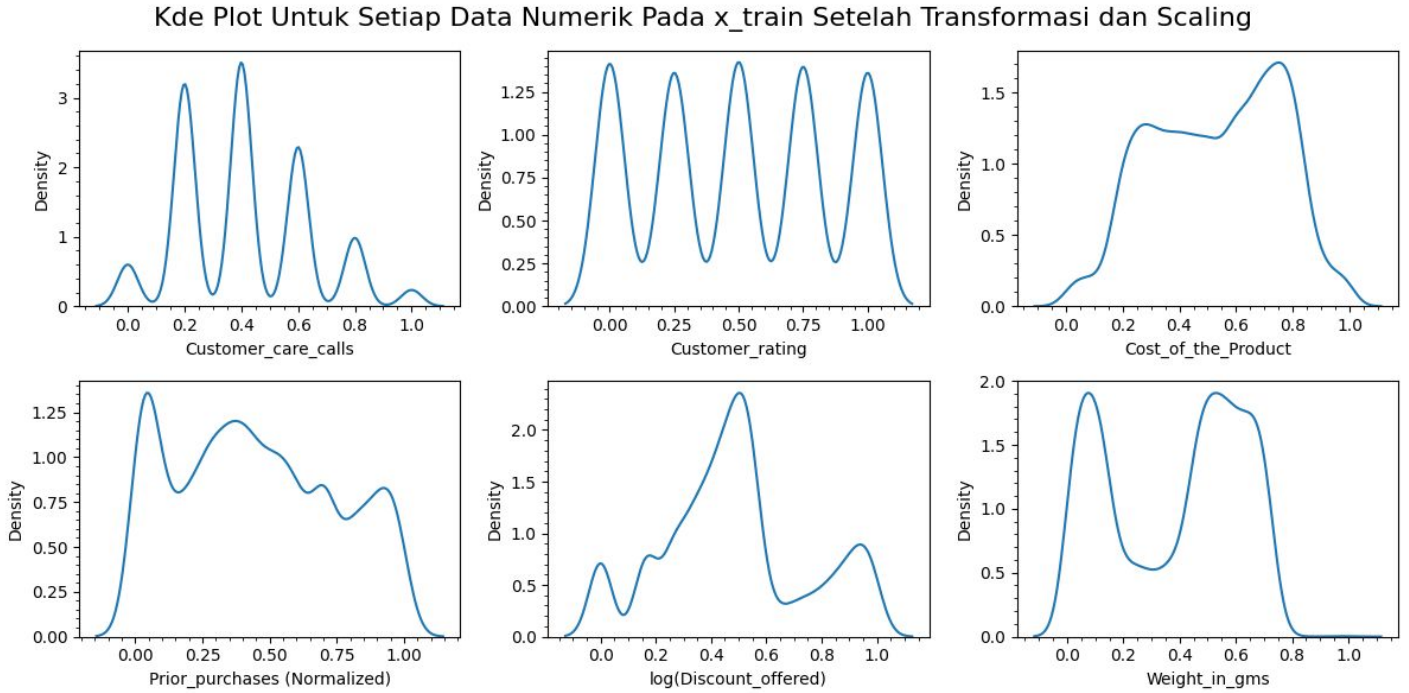
	fitur	derajat skewness
0	Customer_care_calls	0.401
1	Customer_rating	0.002
2	Cost_of_the_Product	-0.161
3	Prior_purchases	1.689
4	Discount_offered	1.781
5	Weight_in_gms	-0.244

Data Cleansing

Feature Transformation

Berikut adalah perbandingan hasil transformasi untuk *Prior_purchases* dan *Discount_offered* menggunakan 2 metode. Sehingga dipilih **Normalizer** untuk *Prior_purchases* dan **Log transformation** untuk *Discount_offered*.





Gambar diatas adalah hasil plotting menggunakan KDEplot pada data numerik setelah transformasi dan scaling menggunakan **MinMaxScaler**. Dan lakukan pemeriksaan *skewness* untuk memastikan bahwa transformasi yang dilakukan berhasil mengurangi skewness pada data. Pemilihan transformasi berikut dapat berubah setelah melihat evaluasi model.

Data Cleansing

Feature Encoding

Setelah melakukan encoding, beberapa kolom kategorikal pada data train diubah menjadi beberapa kolom untuk mempermudah pembentukan model.

1. ***Warehouse_block*** diubah menjadi 4 kolom menggunakan One-Hot Encoding
2. ***Mode_of_Shipment*** diubah menjadi 2 kolom menggunakan One-Hot Encoding
3. ***gender*** diubah menjadi kolom binary, dimana 0 = Female dan 1 = Male
4. ***product_importance*** diubah menjadi kolom ordinal, dimana 0 = low, 1 = medium, dan 2 = high

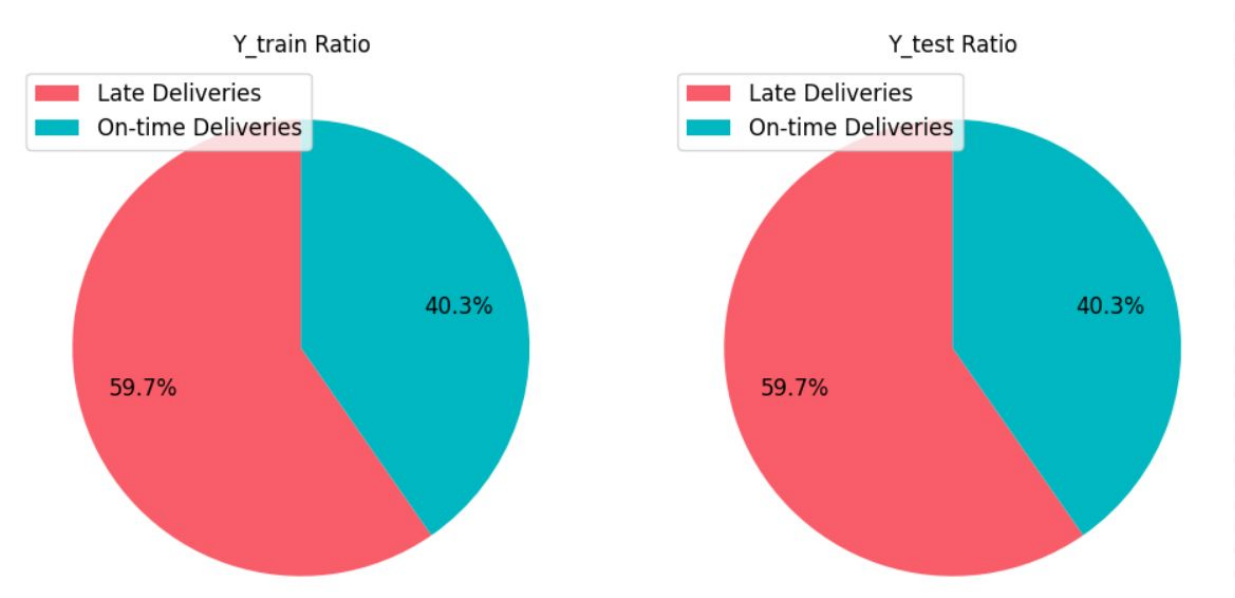
Encoding yang sama kemudian juga akan diterapkan pada data test.

```
x_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8799 entries, 2068 to 10230
Data columns (total 18 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Warehouse_block         8799 non-null   object 
 1   Mode_of_Shipment        8799 non-null   object 
 2   Customer_care_calls     8799 non-null   float64
 3   Customer_rating         8799 non-null   float64
 4   Cost_of_the_Product     8799 non-null   float64
 5   Prior_purchases         8799 non-null   float64
 6   Product_importance      8799 non-null   object 
 7   Gender                  8799 non-null   object 
 8   Discount_offered        8799 non-null   float64
 9   Weight_in_gms           8799 non-null   float64
 10  Warehouse_block_A       8799 non-null   uint8  
 11  Warehouse_block_B       8799 non-null   uint8  
 12  Warehouse_block_C       8799 non-null   uint8  
 13  Warehouse_block_D       8799 non-null   uint8  
 14  Mode_of_Shipment_Flight 8799 non-null   uint8  
 15  Mode_of_Shipment_Road   8799 non-null   uint8  
 16  gender_map              8799 non-null   int64  
 17  product_importance_map  8799 non-null   int64  
dtypes: float64(6), int64(2), object(4), uint8(6)
memory usage: 1.2+ MB
```

Data Cleansing

Class Imbalance



Pada kedua plot, diperoleh distribusi jumlah produk yang sampai tepat waktu yang sama, yaitu 59.7% barang terlambat dan 40.3% barang tepat waktu. Distribusi yang sama diperoleh karena saat pembagian train-test menggunakan ***stratify*** berdasarkan kolom target tersebut.

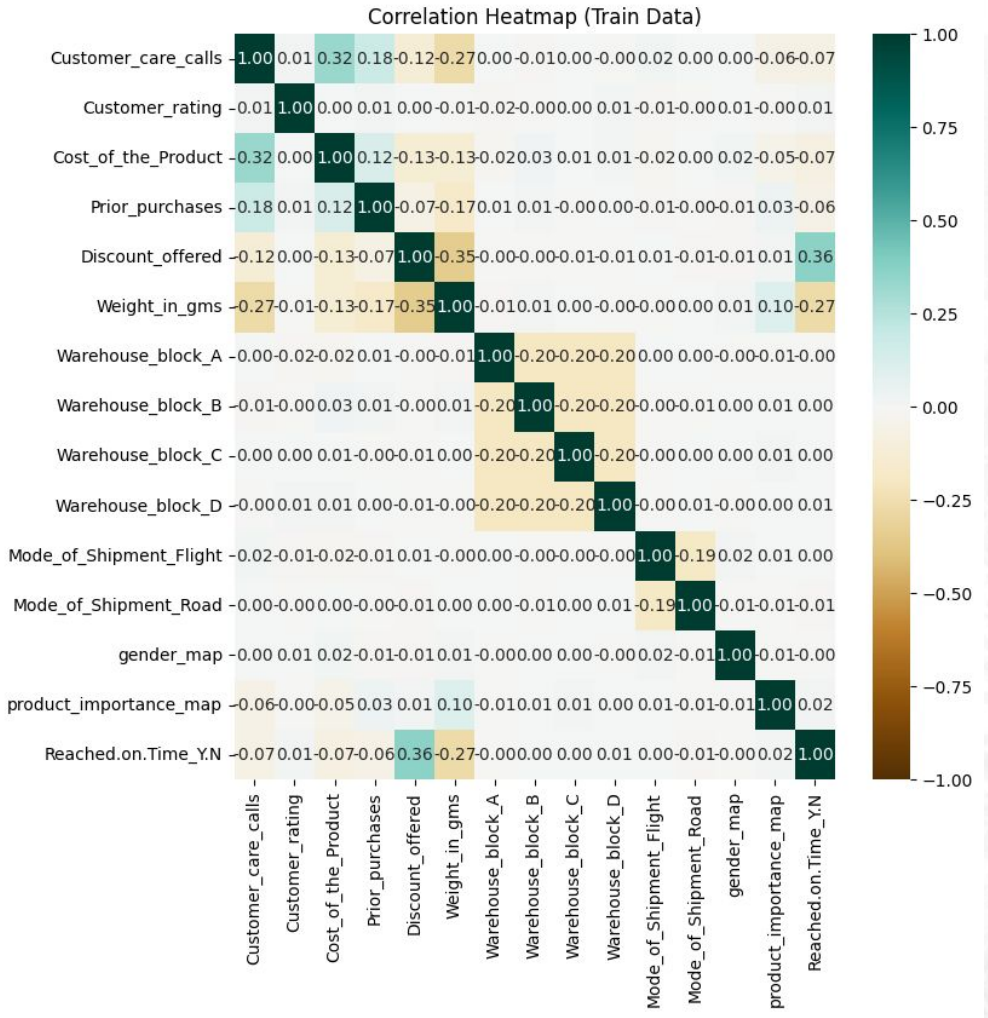
Berdasarkan rasio late vs on-time deliveries pada data train dan data test yang masih mendekati 50:50 (atau tepatnya 60:40), ketidakseimbangan masih dapat diabaikan dan **tidak perlu dilakukan penyeimbangan** lebih lanjut.

Feature Engineering

Feature Selection - Correlation

Berdasarkan korelasi antar fitur pada train data, fitur yang diduga tidak akan berguna pada model adalah ***Customer_rating***, ***warehouse_block*** (semua), ***Mode_of_shipment*** (semua), dan ***gender***, karena korelasinya sangat rendah terhadap semua fitur lainnya

Terdapat beberapa fitur yang memiliki korelasi yang relatif cukup tinggi terhadap fitur lainnya, seperti ***Weight_in_gms*** **dan** ***Discount_offered***, dan juga ***Cost_of_the_product*** **dan** ***Customer_care_calls***, yang memungkinkan terjadinya multikolinearitas jika menggunakan model linear



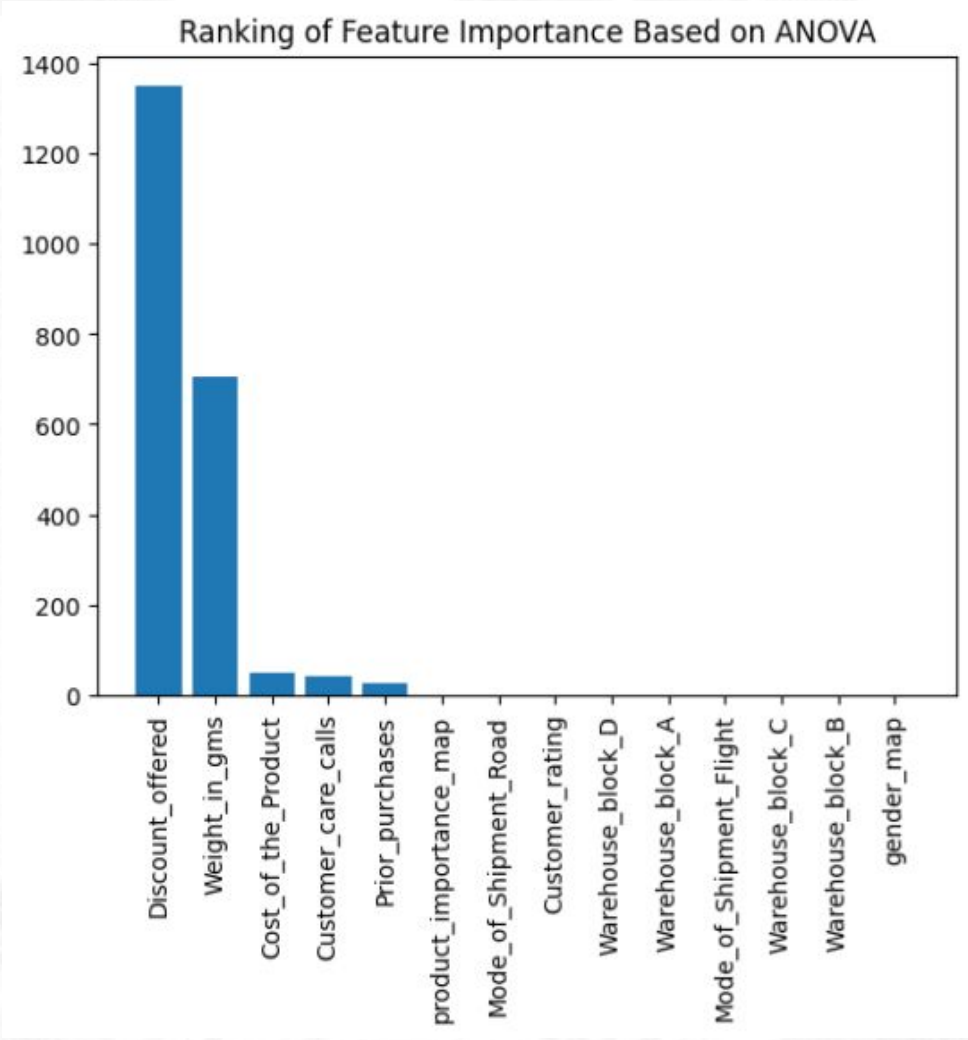
Feature Engineering

Feature Selection - ANOVA

Feature pada data bersifat numerik dan target bersifat kategorik.

Maka untuk melakukan Feature Selection, dapat digunakan metode ANOVA.

Hasil yang diperoleh adalah seperti pada gambar di samping, yaitu berupa ranking Feature Importance. Dari hasil tersebut disimpulkan bahwa fitur yang paling berpengaruh berturut-turut adalah *Discount_offered*, *Weight_in_gms*, *Cost_of_the_product*, *Customer_care_calls*, dan *Prior_purchases*.



Feature Engineering

Feature Extraction

Feature Extraction merupakan tahapan dimana suatu fitur baru dapat dibentuk menggunakan fitur yang ada, umumnya dilakukan pada dua fitur yang memiliki korelasi yang kuat dengan tujuan untuk mengurangi dimensionalitas model dan juga menghindari terjadinya multikolinearitas pada model linear.

Terdapat beberapa feature yang mungkin dapat dilakukan feature extraction :

- Feature *Weight_in_gms* diubah menjadi categorical dengan value (ringan, sedang, berat, sangat berat)
- Feature *Cost_of_the_Product* diubah menjadi categorical dengan value (murah, sedikit mahal, mahal)
- Berdasarkan analisa pada bagian Feature Selection, feature *Weight_in_gms* dan *Discount_offered*, serta *Cost_of_the_product* dan *Customer_care_calls* memiliki korelasi yang cukup besar dan memungkinkan terjadinya multikolinearitas sehingga dapat digabung menjadi satu feature baru, atau salah satu feature dapat di drop.

Feature extraction bersifat opsional dan untuk saat ini masih belum dirasa diperlukan, dan dapat dilakukan jika setelah pembentukan model hasilnya bisa ditingkatkan lagi

Feature Engineering

Additional Features

Beberapa fitur tambahan yang mungkin dapat membantu pembentukan model yang lebih baik adalah sebagai berikut:

- Waktu Keterlambatan -> supaya keterlambatan lebih terukur secara durasi, tidak hanya terlambat atau tidak
- Tanggal Pengiriman -> untuk memeriksa apakah ada hubungan keterlambatan dengan hari/tanggal tertentu, misal saat libur atau hari raya atau saat hari sibuk/padat pengiriman
- Wilayah Pengiriman -> apakah keterlambatan tergantung kepada wilayah pengiriman?
- Jenis/Jasa Kurir -> apakah keterlambatan tersebar secara merata, atau memiliki korelasi yang kuat terhadap kurir yang digunakan untuk pengiriman barang?
- Jenis promo -> apakah ada promo yang diberikan oleh e-commerce kepada customer, misal berupa *free /discount* ongkir.

Git

Link Repository Github: https://github.com/mezkymy/ecommerce-ds

Rakamin Academy

STAGE 3
Modeling

Step Data Modeling

Split Data Train & Test

Sebelum melakukan modeling, lakukan split data train dan test supaya pemrosesan data dapat dilakukan hanya kepada data *train* saja terlebih dahulu untuk mencegah terjadinya *data leakage*. Setelah itu, transformasi data yang sama yang telah diterapkan pada data *train* dapat digunakan pada data *test*.

Tahapan ini **sudah dilakukan pada stage 2,** karena pada tahapan preprocessing removal outlier dilakukan hanya pada data train sehingga split data perlu dilakukan terlebih dahulu sebelum pemrosesan.

Step Data Modeling

Modeling

Data yang tersedia berupa data dengan label yang bersifat kategorik (0 dan 1), sehingga pada bagian modeling dipilih metode-metode dari supervised learning berjenis klasifikasi, dimana tujuan dari model yang dibentuk adalah untuk memprediksi apakah suatu pengiriman akan mengalami keterlambatan (nilai 0) atau tidak (nilai 1). Model-model yang digunakan pada studi kasus ini adalah:

1. Decision Tree
2. Random Forest
3. Logistic Regression
4. K-Nearest Neighbor
5. XGBoost

Step Data Modeling

Modeling

1. **Decision Tree**
   Decision tree merupakan algoritma machine learning yang menyajikan algoritma dengan pernyataan bersyarat, yang meliputi cabang untuk mewakili langkah-langkah pengambilan keputusan yang dapat mengarah pada hasil yang menguntungkan.

2. **Random Forest**
   Random Forest merupakan algoritma machine learning yang membangun beberapa decision tree dan menggabungkannya untuk mendapatkan prediksi yang lebih akurat dan stabil.

3. **Logistic Regression**
   Logistic Regression adalah sebuah algoritma klasifikasi untuk mencari hubungan antara fitur (input) diskrit/kontinu dengan probabilitas hasil output diskrit tertentu.

4. **K-Nearest Neighbor**
   KNN merupakan algoritma klasifikasi yang paling sederhana dalam mengklasifikasikan sebuah gambar kedalam sebuah label. Metode ini mengklasifikasikan berdasarkan jarak terdekat dengan objek lain (tetangga).

5. **XGBoost**
   XGBoost (Extreme Gradient Boosting) adalah algoritma boosting tree yang digunakan untuk tugas-tugas klasifikasi dan regresi. Algoritma ini menggabungkan beberapa pohon keputusan sederhana untuk membuat model yang lebih kompleks dan akurat.

Step Data Modeling

Model Evaluation: Pemilihan dan perhitungan metrics model

1. ***Accuracy***
   Metrics Akurasi biasa digunakan ketika masing-masing label mempunyai kepentingan yang sama dan jumlah labelnya seimbang.

2. ***Precision***
   Metrics *Precision* biasa digunakan ketika kita lebih memperhatikan jumlah False Positive (FP) yang sebaiknya lebih sedikit dengan label yang seimbang.

3. ***Recall***
   Metrics *Recall* digunakan jika kita tidak memperbolehkan nilai False Negative yang besar dan labelnya seimbang.

Dari ketiga metrics diatas, metrics akurasi/*accuracy* dinilai paling cocok digunakan untuk evaluasi model yang dibentuk, karena akurasi menghitung proporsi keseluruhan prediksi yang benar dari total jumlah prediksi. Dalam hal ini, akurasi akan memberikan gambaran tentang seberapa baik model dapat memprediksi dengan benar pada keseluruhan data.

Step Data Modeling

Model Evaluation: Apakah model sudah best-fit?

Pada step ini, dilakukan cross-validation dengan menggunakan 10 fold untuk masing-masing model. Tujuannya adalah untuk membandingkan score accuracy dari data training dan score hasil cross validation antar model. Berikut adalah hasil yang diperoleh:

	Model	Training Accuracy	CV Accuracy (mean)	CV Accuracy (std)	Training Precision	CV Precision (mean)	CV Precision (std)	Training Recall	CV Recall (mean)	CV Recall (std)
0	Decision Tree	1.000000	0.645931	0.014312	1.000000	0.700896	0.011990	1.000000	0.710254	0.024738
1	Random Forest	1.000000	0.653100	0.015333	1.000000	0.748967	0.013727	1.000000	0.630266	0.022305
2	Logistic Regression	0.654255	0.651712	0.012234	0.731219	0.728994	0.013486	0.665505	0.663774	0.022389
3	KNN	0.775671	0.645933	0.015892	0.834822	0.717377	0.014799	0.778230	0.671906	0.024164
4	XGBoost	0.891304	0.651248	0.014945	0.949925	0.734680	0.011710	0.863451	0.650985	0.024796

Dari hasil tersebut, model yang dipilih adalah **Random Forest** karena memiliki rata-rata score accuracy cross validation yang paling tinggi dan standar deviasi score yang relatif rendah. Namun, jika diperhatikan model Random Forest menghasilkan score Training Accuracy yang bernilai 1, maka model mengalami overfitting sehingga perlu dilakukan step lebih lanjut yaitu hyperparameter tuning.

Step Data Modeling

Hyperparameter Tuning

Setelah ditentukan bahwa model yang akan digunakan adalah Random Forest, kemudian hyperparameter yang akan digunakan dalam membentuk model akan di-*tuning* supaya model yang dihasilkan lebih baik, terutama dari segi fitting terhadap data train untuk mencegah terjadinya overfitting maupun underfitting.

```
param_grid = {
    'classifier__n_estimators' : [100, 200],
    'classifier__criterion' : ['gini', 'mse', 'entropy'],
    'classifier__max_depth' : [None, 5, 10, 100],
    # 'classifier__min_samples_split' : [2, 10], #optional
    # 'classifier__min_samples_leaf' : [1, 5], #optional
    'classifier__max_features' : ['sqrt', 'log2', None, 'auto'],
    # 'classifier__bootstrap' :[True, False], #optional

}
```

(contoh kombinasi hyperparameter yang akan diuji)

Adapun kombinasi dari hyperparameter diuji menggunakan randomized search (untuk jumlah kombinasi yang banyak) ataupun grid search (untuk jumlah kombinasi yang relatif sedikit) untuk mencari kombinasi mana yang menghasilkan model terbaik. Melalui proses ini, ditentukan parameter yang terbaik untuk membentuk model Random Forest pada studi kasus ini sebagai berikut:

n_estimators: 300,
max_features: None,
max_depth: 5,
criterion: 'entropy'

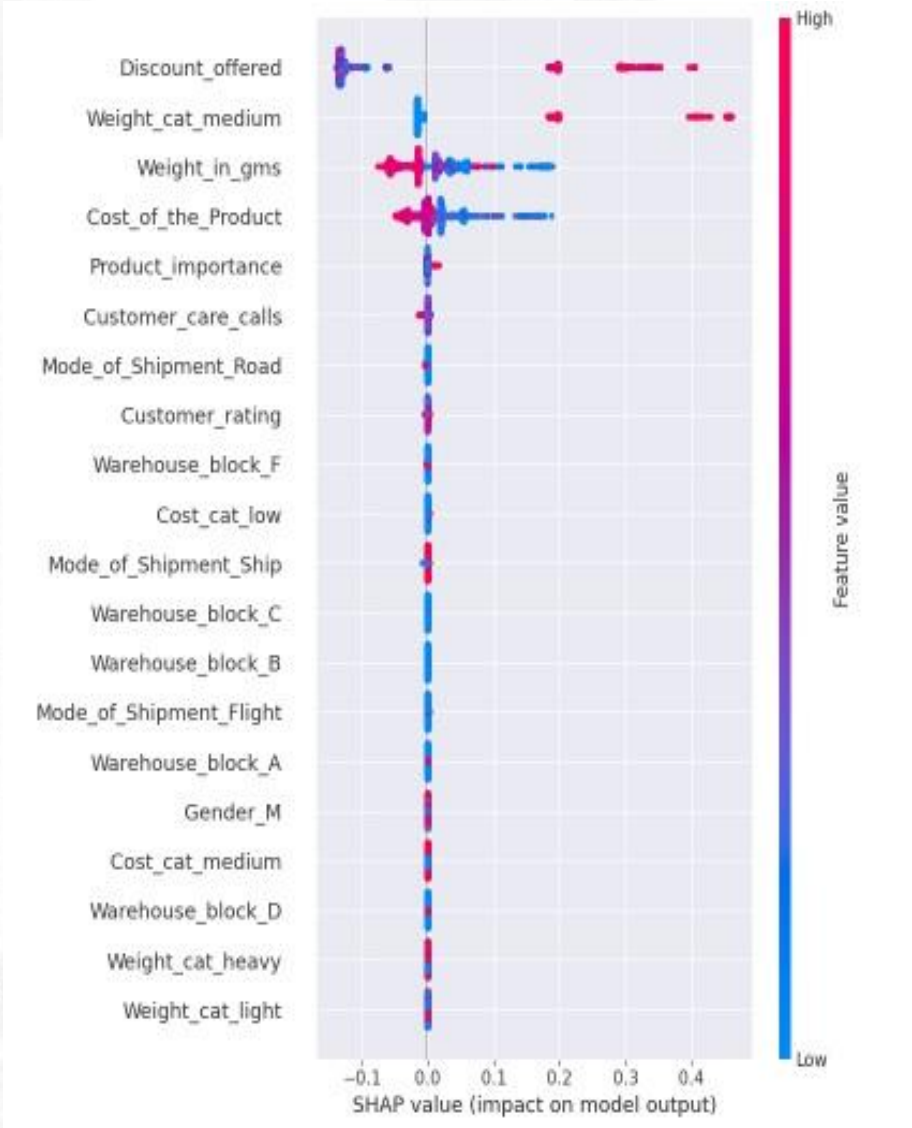
Kombinasi hyperparameter di atas menghasilkan model dengan Accuracy score data **TRAIN** sebesar **69.149 %** dan Accuracy score data **TEST** sebesar **67.545 %,** sehingga model dapat dikatakan tidak underfit maupun overfit secara signifikan

Feature Importance

SHAP

Interpretasi:

- Fitur **Discount_offered** menjadi fitur paling penting terhadap target dan berkorelasi secara positif. Semakin tinggi nilai discount offered, maka kecenderungan fitur untuk memprediksi targetnya semakin berpengaruh besar.
- Fitur **Weight_cat_medium** menjadi fitur kedua yang terpenting terhadap target dan berkorelasi secara positif. Semakin tinggi nilai Weight_cat_medium, maka kecenderungan fitur untuk memprediksi targetnya semakin berpengaruh besar.
- Fitur **Weight_in_gms** menjadi fitur ketiga yang terpenting terhadap target namun berkorelasi secara negatif. Semakin rendah nilai Weight_in_gms, maka kecenderungan fitur untuk memprediksi targetnya semakin berpengaruh besar.
- Fitur **Cost_of_the_Product** menjadi fitur keempat yang terpenting terhadap target namun berkorelasi secara negatif. Semakin rendah nilai Cost_of_the_Product , maka kecenderungan fitur untuk memprediksi targetnya semakin berpengaruh besar.



Business Insights

Berdasarkan hasil evaluasi feature menggunakan explainable AI dengan SHAP, dapat dilihat bahwa:

1. fitur '**Discount_offered**' memiliki pengaruh yang besar terhadap ketepatan waktu pengiriman barang.
2. fitur "**Weight_in_gms**" memiliki pengaruh yang cukup signifikan terhadap hasil prediksi. Semakin besar nilai fitur "Weight_in_gms", semakin besar kemungkinan bahwa pengiriman akan ditolak oleh pelanggan.
3. fitur "**Warehouse_block**" (A, B, C, D, F), "**Mode_of_Shipment**" (Road, Ship, Flight), "**Customer_care_calls**", dan " **Customer_rating**" secara umum tidak memiliki pengaruh / pengaruh sangat kecil kepada keterlambatan pengiriman barang. Artinya perusahaan tidak perlu memikirkan tipe barang, metode pengiriman, panggilan / complain yang masuk, dan rating dari barang sebagai hal yang dapat mengakibatkan keterlambatan.
4. fitur "**Cost_of_Product**" memiliki pengaruh yang signifikan, semakin tinggi nilainya maka semakin rendah kemungkinan sebuah transaksi akan dikategorikan sebagai kecurangan.

Business Insights

Selain dari Business Insight yang bisa ditarik dari Feature Importance, terdapat beberapa Business Insight penting yang dapat ditarik berdasarkan temuan pada tahapan EDA:

1. Keterlambatan pengiriman barang hanya terjadi pada barang yang beratnya ada di kategori *light*/ringan (1000-2000 gram) dan *heavy*/berat (4000-6000 gram)
2. Keterlambatan pengiriman barang hanya terjadi pada barang yang memiliki nilai diskon yang ditawarkan sebesar 10% atau lebih kecil

Business Insights

Rekomendasi

1. Memberikan penawaran diskon yang menarik berdasarkan data historis **Discount_offered** Dengan begitu, pelanggan yang pernah melakukan pembelian sebelumnya cenderung lebih tertarik untuk melakukan pembelian lagi, dan peluang konversi dapat ditingkatkan.
2. Mengoptimalkan strategi pemasaran dengan mempertimbangkan variabel **Discount_offered**. Misalnya, menargetkan kampanye pemasaran kepada pelanggan dengan memberikan diskon yang sesuai dengan karakteristik pelanggan tersebut.
3. Meningkatkan kualitas data terkait **Discount_offered** dengan melakukan verifikasi dan validasi data secara teratur.
4. Memastikan bahwa paket yang dikirim tidak melebihi batas berat yang diizinkan atau mengevaluasi ulang kebijakan pengiriman untuk produk-produk tertentu yang cenderung memiliki berat yang lebih besar terkait **Weight_in_gms**
5. Memastikan bahwa **Cost_of_Product** atau biaya produksi dapat dikelola dengan baik untuk mengoptimalkan profit per unit produk. Hal ini juga dapat membantu mengurangi kemungkinan terjadinya fraud dalam transaksi , sehingga dapat meningkatkan kepercayaan pelanggan dan reputasi perusahaan.

Business Insights

Rekomendasi

Rekomendasi Berdasarkan Business Insight dari Tahapan EDA:

1. Untuk barang yang beratnya ada di kategori ringan (1000-2000 gram) dan berat (4000-6000 gram), perlu perhatian khusus karena seluruh keterlambatan dalam dataset yang tersedia hanya terjadi pada kategori berat tersebut
2. Untuk barang yang memiliki nilai diskon yang ditawarkan sebesar 10% atau lebih kecil perlu perhatian khusus karena seluruh keterlambatan dalam dataset yang tersedia hanya terjadi pada kategori tersebut.
3. Perlu dilakukan pengumpulan data yang lebih detil dengan jumlah fitur yang lebih banyak untuk melihat korelasi yang lebih konkrit antara berat dan nilai diskon terhadap keterlambatan pengiriman; misal apakah barang dengan nilai diskon dan berat tertentu dikirimkan menggunakan jasa pengiriman yang sama, atau apa keterlambatan pengiriman terjadi pada jenis-jenis barang yang lebih spesifik