KELOMPOK 2

Stage 2

Mentor : Mas Mirza

Elvis Muh. Rizqy

Fuji Resti M

Ni Kadek Yulia Cyntia Dewi

Haolia

Luthfi Adnan Rahmantyo

Pre processing

Pada tahap ini kita membersihkan data yang kotor, sehingga nantinya data yang telah dibersihkan dalam penerapan model Machine Learning akan membuat kinerja yang semakin baik.

	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
0	1	D	Flight	4	2	177	3	low	F	44	1233	1
1	2	F	Flight	4	5	216	2	low	M	59	3088	1
2	3	A	Flight	2	2	183	4	low	M	48	3374	1
3	4	B	Flight	3	3	176	4	medium	M	10	1177	1
4	5	C	Flight	2	2	184	3	medium	F	46	2484	1

A. Handle missing values

```
df_new.isna().sum()
```

```
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         0
dtype: int64
```

- Tidak ada nilai yang hilang sehingga tidak ada yang harus di handle

B. Handle Duplicated data

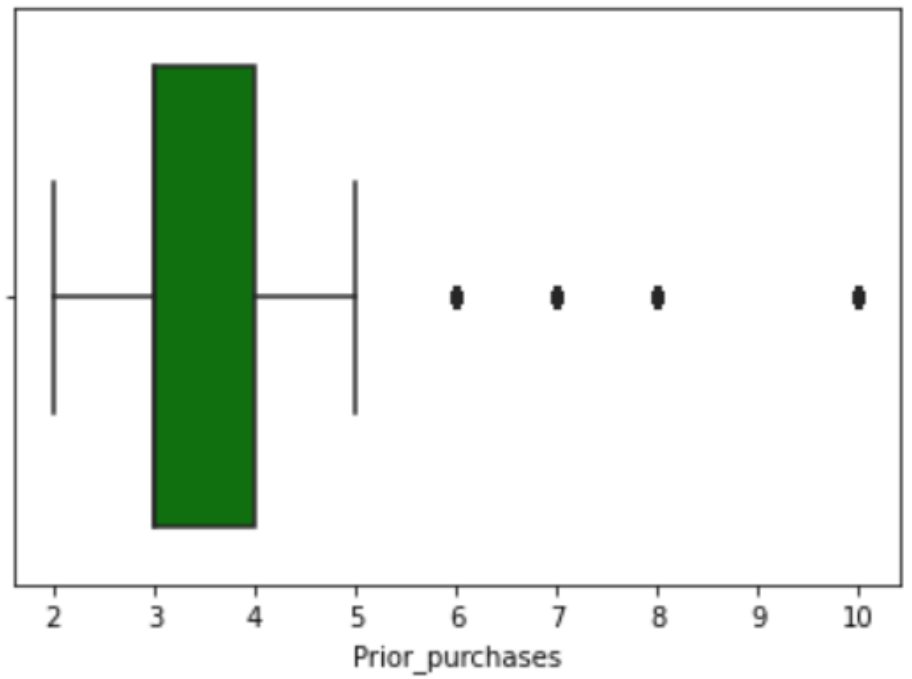
```
df_new.duplicated().sum()
```

```
0
```

Tidak ada values yang duplikasi sehingga tidak ada data yang harus di handle

C. Handle outliers

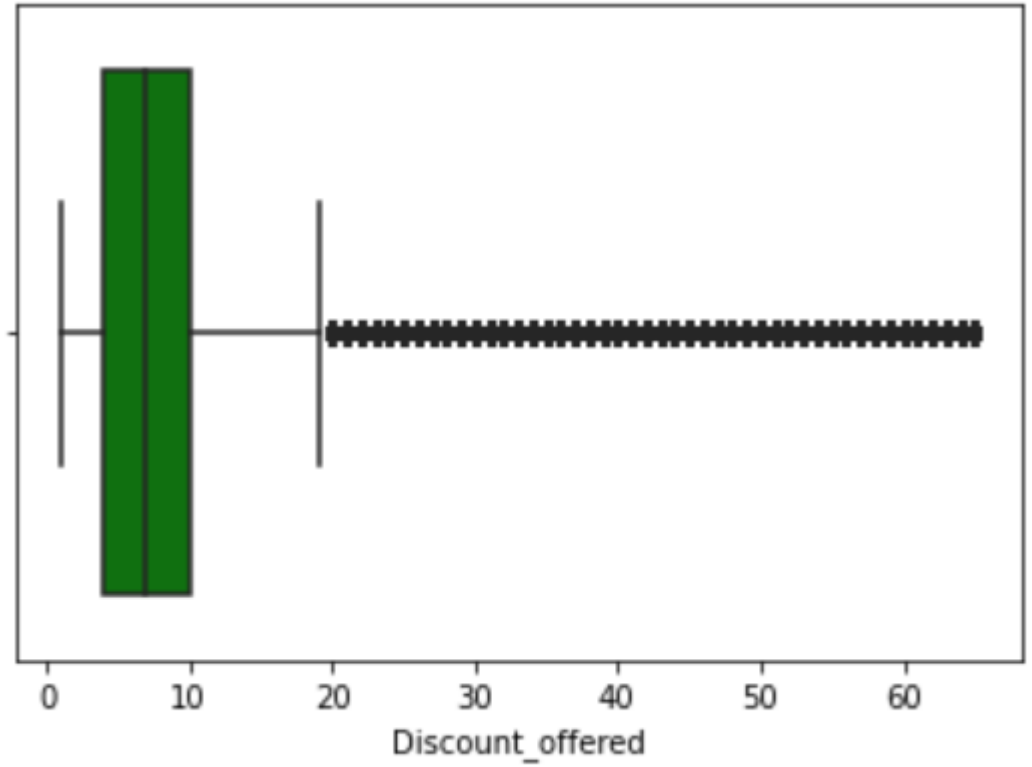
```
sns.boxplot(x=df_new["Prior_purchases"], color='green', orient='v');
```



- Outlier pada kolom prior purchases tidak perlu dibuang dikarenakan nilainya masih dalam batas wajar (kecuali ada nilai yang < 0 sehingga harus dilakukan drop pada kolom tersebut)

C. Handle outliers

```
sns.boxplot(x=df_new["Discount_offered"], color='green', orient='v');
```



- Outlier pada kolom Discount offered tidak perlu dibuang dikarenakan discount yang diberikan masih dalam batas wajar (karena discount yang diberikan tidak melebihi 100%)

D. Feature Transformation

Pada tahap ini kita harus melakukan transformasi data pada grafik yang ada ke dalam distribusi normal

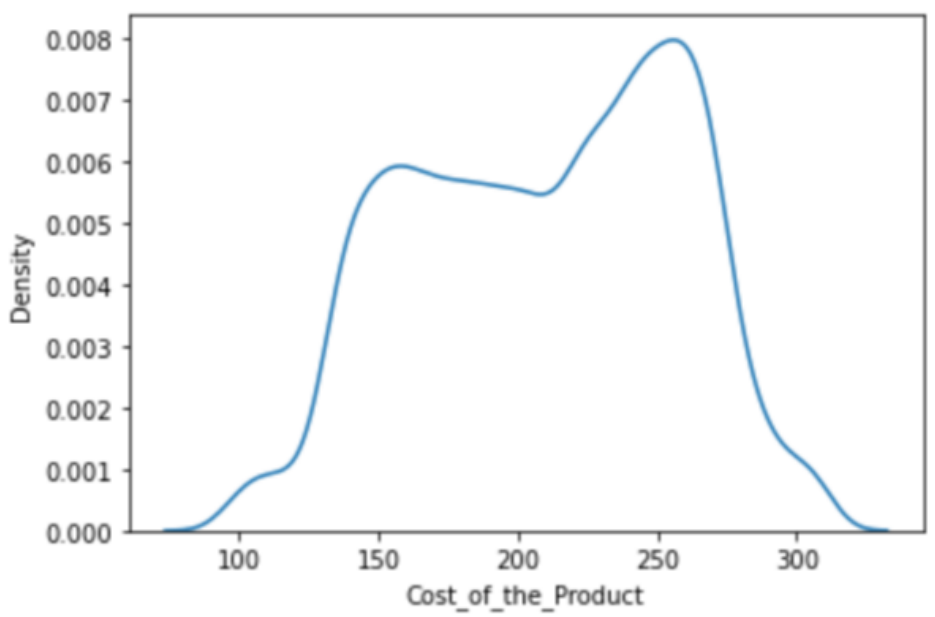
Selain itu, kita juga harus melihat seberapa skewed data yang kita miliki terlebih dahulu:

- jika nilai skewed -0.5 atau 0.5 adalah Fairy Symmetrical
- jika nilai skewed -0.5 sampai -1.0 dan 0.5 sampai 1.0 adalah Moderate Skewed
- jika nilai skewed < -1.0 dan > 1.0 adalah Highly Skewed

D. Feature Transformation

Cost of Product

```
sns.kdeplot(x=df_new["Cost_of_the_Product"]);
```



- Grafik Cost of Product termasuk ke dalam Bimodal

- Menggunakan fungsi transpose untuk melihat nilai skewednya, sehingga kita dapat menentukan penggunaan metode perubahan distribusi yang ada

```
df_new[('Cost_of_the_Product')].agg(['skew', 'kurtosis']).transpose().reset_index()
```

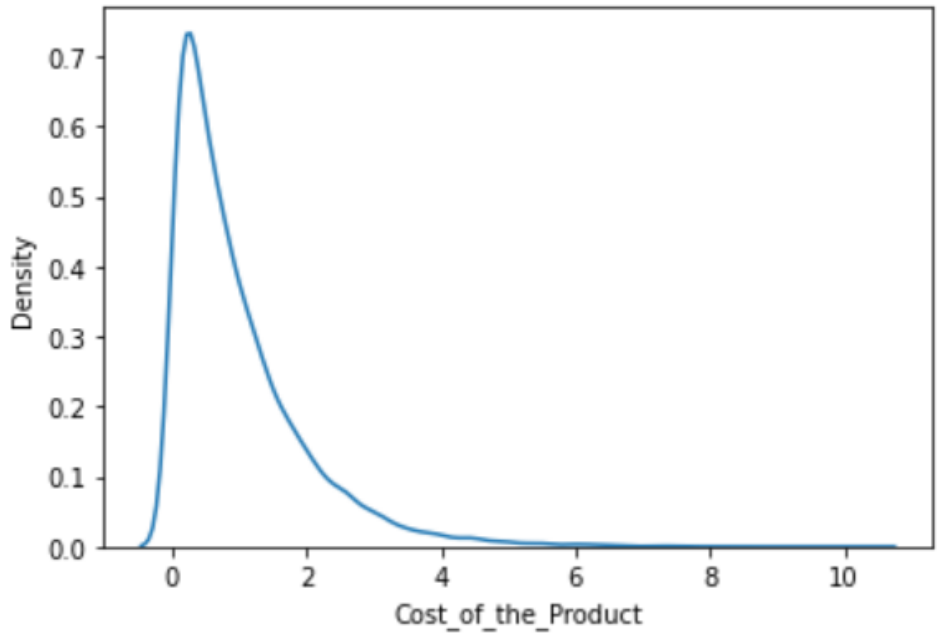
	index	Cost_of_the_Product
0	skew	-0.157117
1	kurtosis	-0.972160

D. Feature Transformation

Pertama-tama kita menggunakan metode exponential untuk mengubah grafik bimodal ke eksponensial. Kemudian menggunakan metode Box cox untuk merubah distribusinya ke distribusi normal

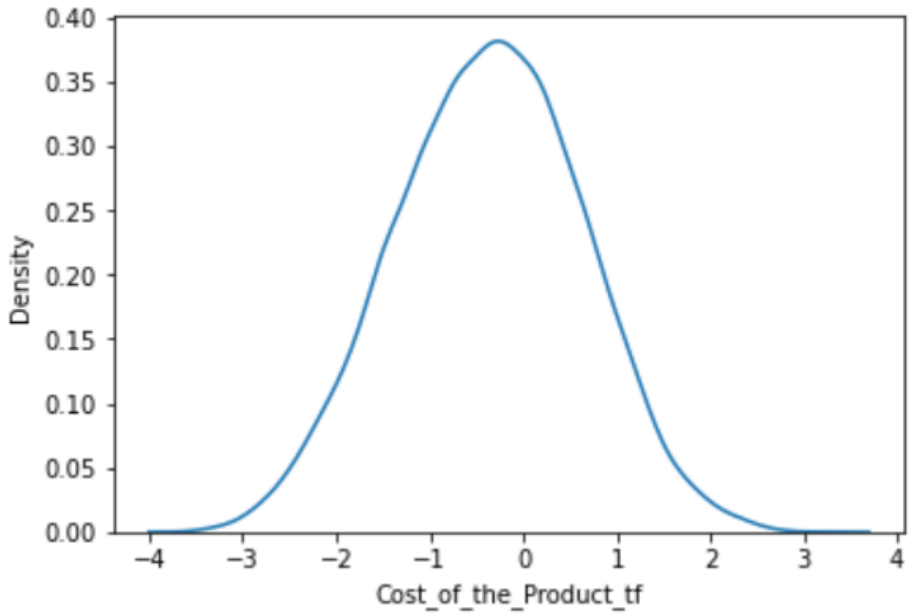
```
df_new['Cost_of_the_Product'] = np.random.exponential(size=10999)
```

```
sns.distplot(df_new['Cost_of_the_Product'], hist=False, kde=True);
```



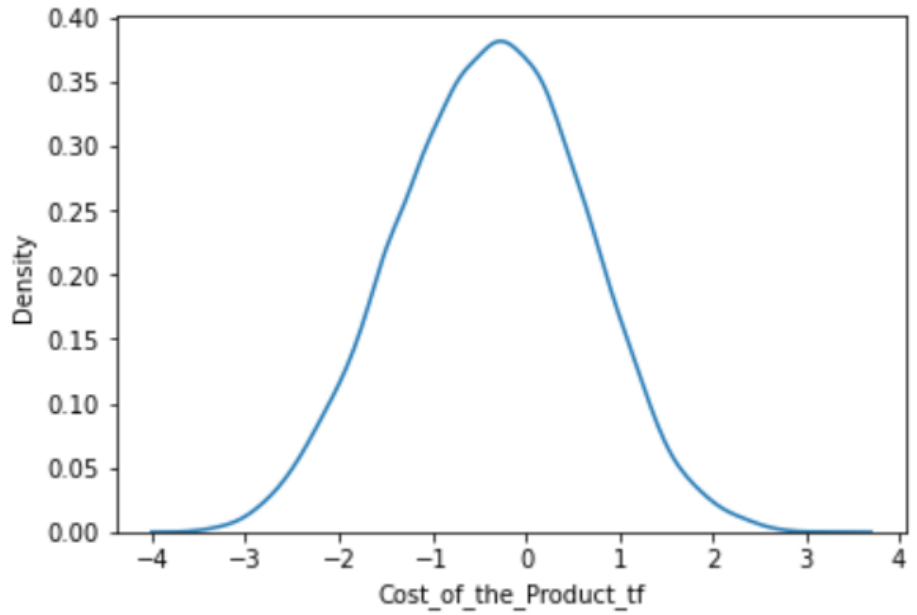

```
df_new['Cost_of_the_Product_tf'], best_lambda = boxcox(df_new['Cost_of_the_Product'])
```

```
sns.kdeplot(x=df_new["Cost_of_the_Product_tf"]);
```



D. Feature Transformation

Kemudian setelah menjadi distribusi normal, kita melakukan standarisasi dengan fitur StandardScaler



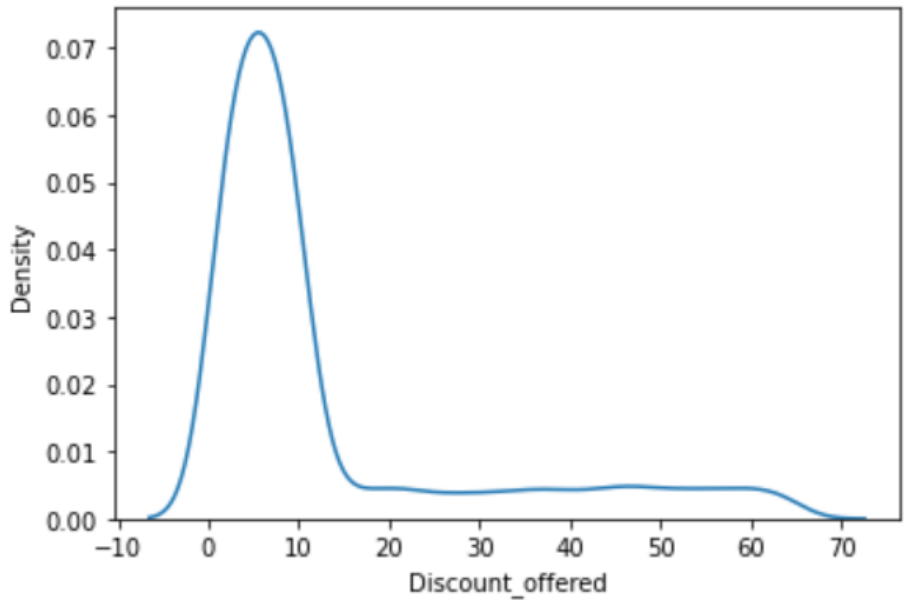

```
df_new['Cost_of_the_Product_tf'] = StandardScaler().fit_transform(df_new['Cost_of_the_Product'].values.reshape(len(df), 1))
```

D. Feature Transformation

Rakamin Academy

Discount Offered

```
sns.kdeplot(x=df_new["Discount_offered"]);
```



- Menggunakan fungsi transpose untuk melihat nilai skewednya, sehingga kita dapat menentukan penggunaan metode perubahan distribusi yang ada

```
df_new[('Discount_offered')].agg(['skew', 'kurtosis']).transpose().reset_index()
```

	index	Discount_offered
0	skew	1.798929
1	kurtosis	2.000586

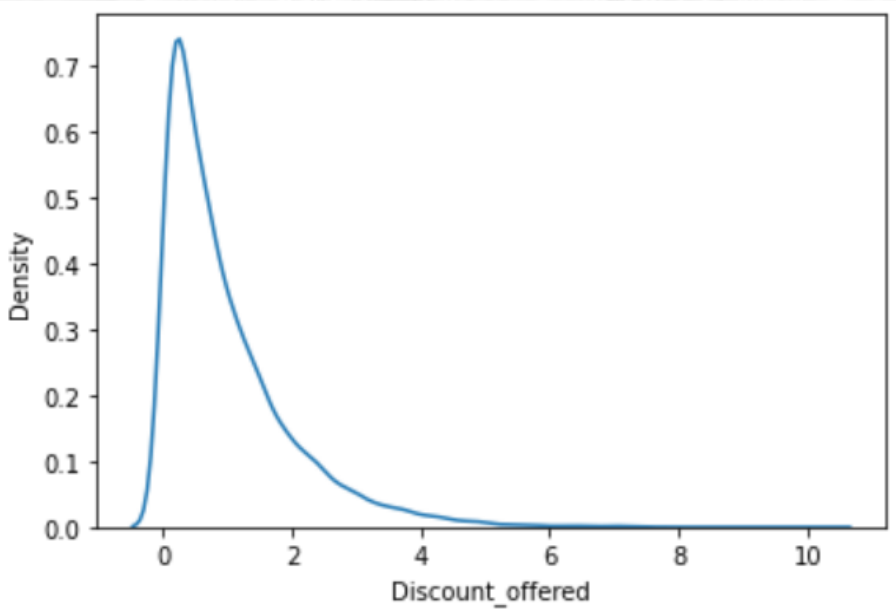
- Grafik Discount Offered termasuk ke dalam Positively skewed atau condong ke kanan

D. Feature Transformation

Pada awalnya setelah menggunakan metode Log Transformasi karena skewed ke kanan, distribusi yang muncul berubah menjadi bimodal, maka dari itu metode yang paling cocok yaitu menggunakan metode eksponensial. Kemudian menggunakan metode Box cox untuk merubah distribusinya ke distribusi normal

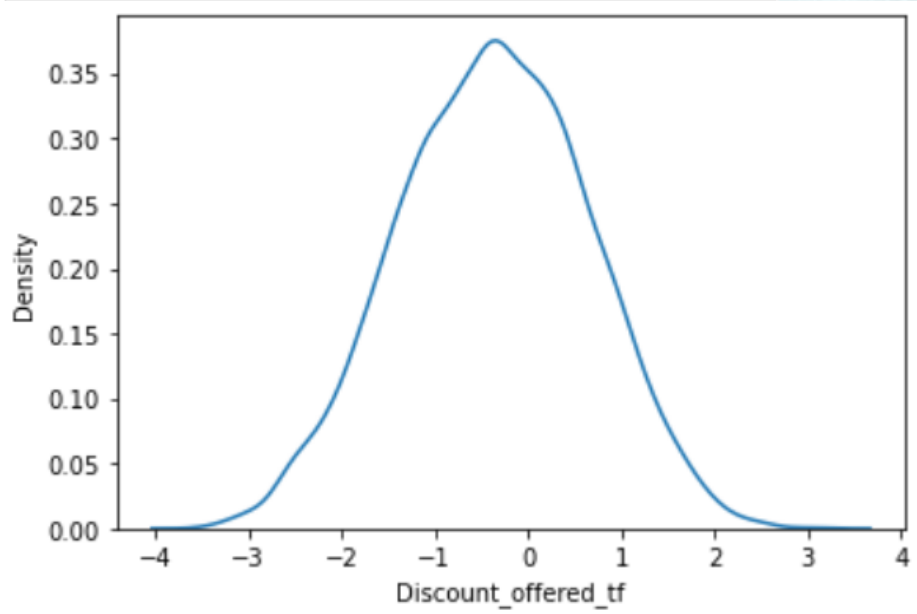
```
df_new['Discount_offered'] = np.random.exponential(size=10999)
```

```
sns.distplot(df_new['Discount_offered'], hist=False, kde=True);
```



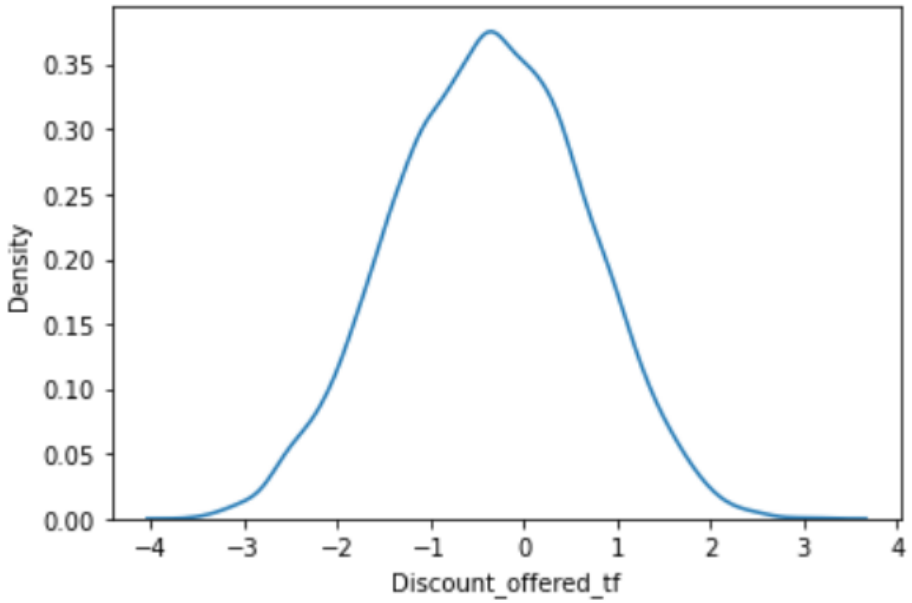

```
df_new['Discount_offered_tf'], _ = boxcox(df_new['Discount_offered'])
```

```
sns.kdeplot(x=df_new["Discount_offered_tf"]);
```



D. Feature Transformation

Kemudian setelah menjadi distribusi normal, kita melakukan standarisasi dengan fitur StandardScaler

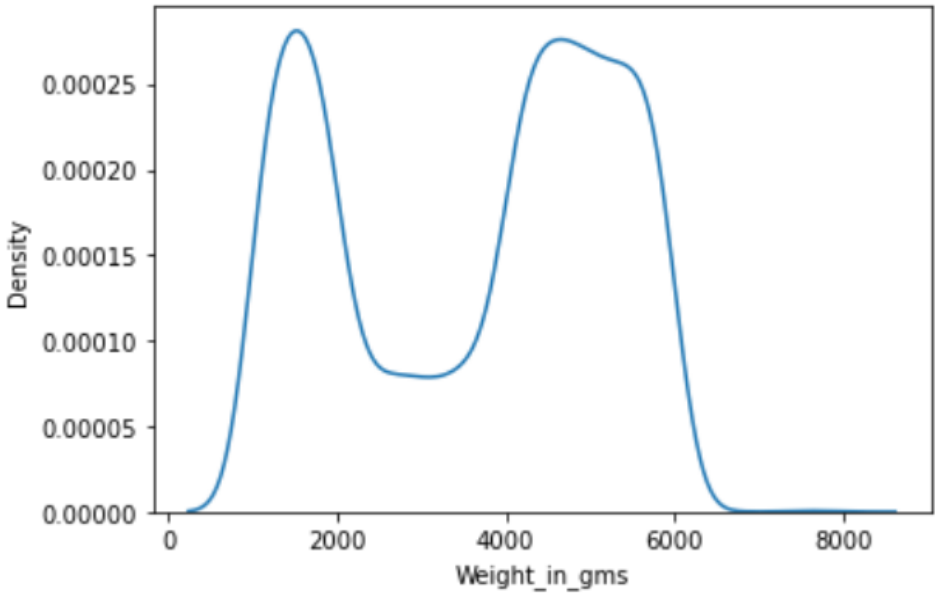



```
df_new['Discount_offered_tf'] = StandardScaler().fit_transform(df_new['Discount_offered'].values.reshape(len(df), 1))
```

D. Feature Transformation

Weight in gms

```
sns.kdeplot(x=df_new['Weight_in_gms']);
```



- Menggunakan fungsi transpose untuk melihat nilai skewednya

```
df_new[('Weight_in_gms')].agg(['skew', 'kurtosis']).transpose().reset_index()
```

	index	Weight_in_gms
0	skew	-0.249747
1	kurtosis	-1.447671

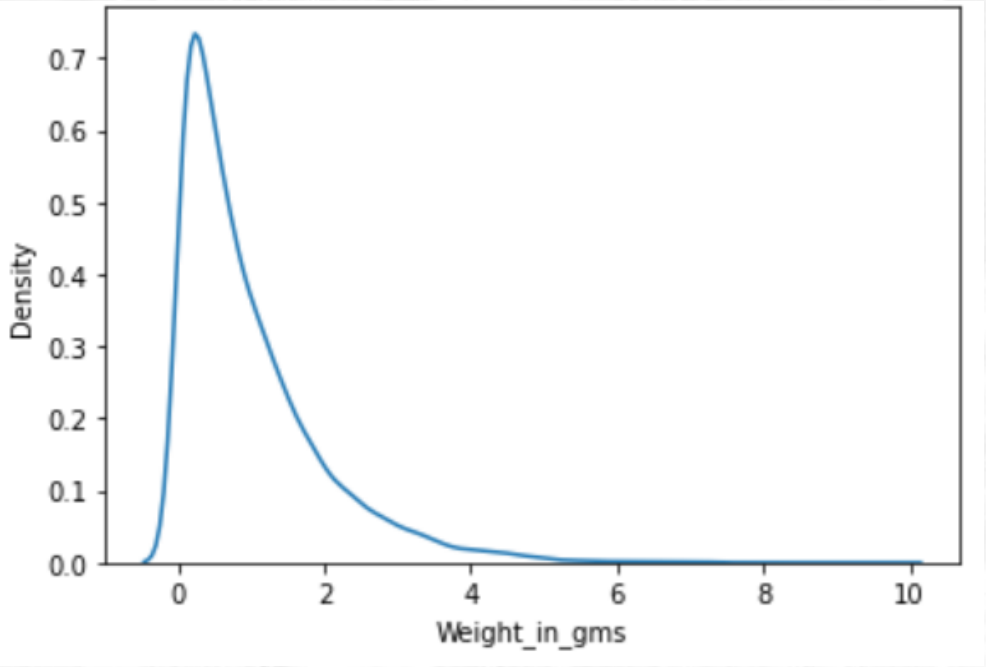
- Grafik Weight in gms termasuk ke dalam Bimodal

D. Feature Transformation

Pertama-tama kita menggunakan metode exponential untuk mengubah grafik bimodal ke eksponensial. Kemudian menggunakan metode Box cox untuk merubah distribusinya ke distribusi normal

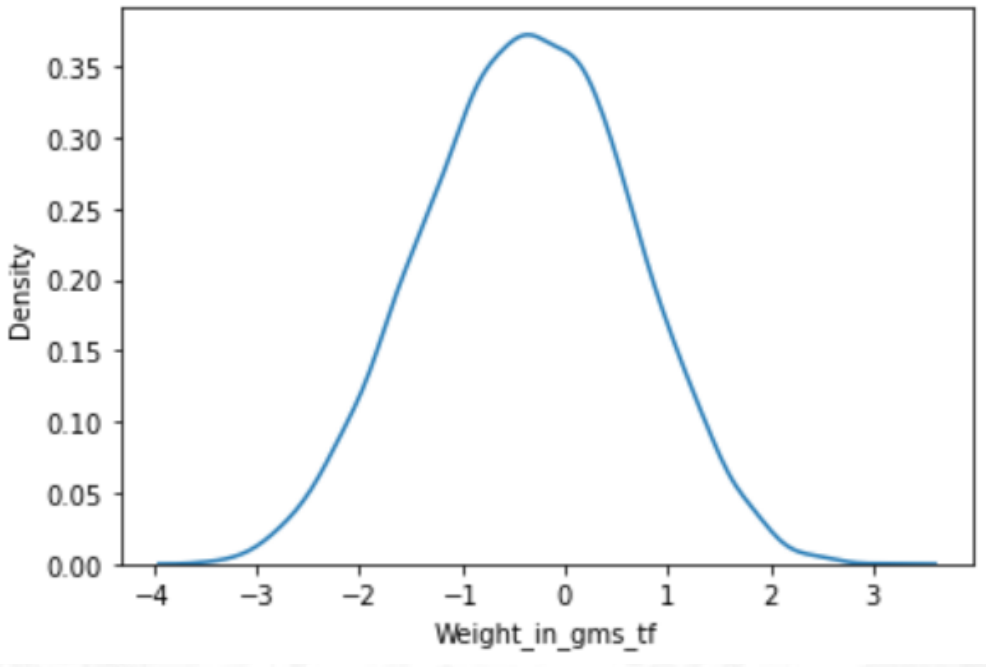
```
df_new['Weight_in_gms'] = np.random.exponential(size=10999)
```

```
sns.distplot(df_new['Weight_in_gms'], hist=False, kde=True);
```



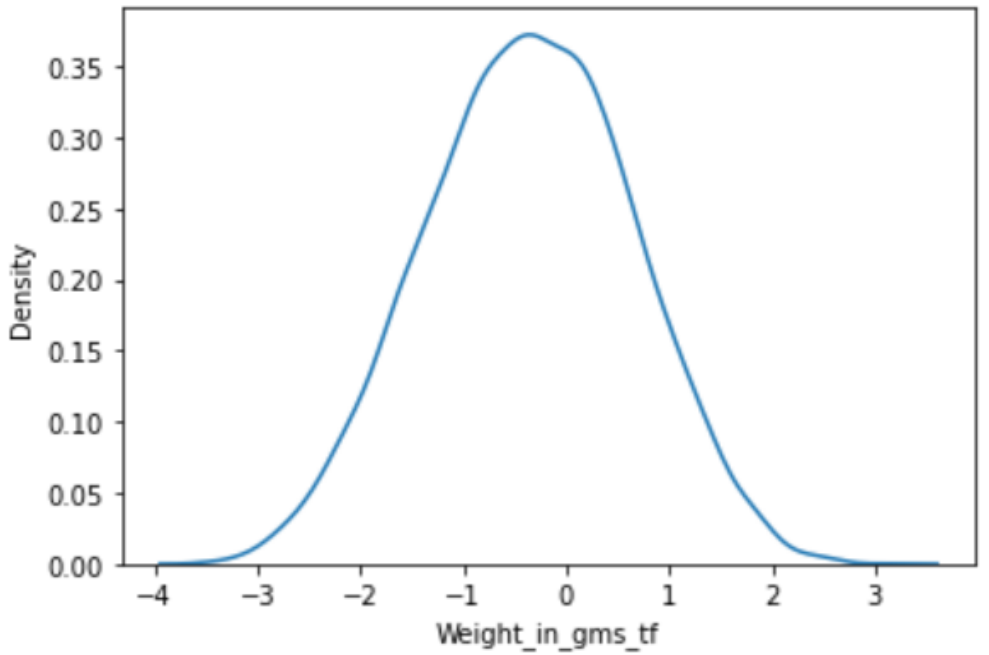

```
df_new['Weight_in_gms_tf'], _ = boxcox(df_new['Weight_in_gms'])
```

```
sns.kdeplot(x=df_new['Weight_in_gms_tf']);
```



D. Feature Transformation

Kemudian setelah menjadi distribusi normal, kita melakukan standarisasi dengan fitur StandardScaler



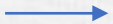

```
df_new['Weight_in_gms_tf'] = StandardScaler().fit_transform(df['Weight_in_gms'].values.reshape(len(df), 1))
```

D. Feature Transformation

Setelah semua data telah selesai distandarisasi kemudian, kita melakukan pengecekan kolom-kolom di dataframe kita yang telah diupdate, kemudian menghapus kolom yang lama

```
df_new.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10999 entries, 0 to 10998
Data columns (total 14 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Warehouse_block        10999 non-null  object
 1   Mode_of_Shipment       10999 non-null  object
 2   Customer_care_calls    10999 non-null  int64
 3   Customer_rating        10999 non-null  int64
 4   Cost_of_the_Product    10999 non-null  float64
 5   Prior_purchases        10999 non-null  int64
 6   Product_importance     10999 non-null  object
 7   Gender                 10999 non-null  object
 8   Discount_offered       10999 non-null  float64
 9   Weight_in_gms          10999 non-null  float64
 10  Reached_on_Time        10999 non-null  int64
 11  Cost_of_the_Product_tf 10999 non-null  float64
 12  Discount_offered_tf    10999 non-null  float64
 13  Weight_in_gms_tf       10999 non-null  float64
dtypes: float64(6), int64(4), object(4)
memory usage: 1.2+ MB
```



```
df_new = df_new.drop(columns=['Discount_offered', 'Cost_of_the_Product','Weight_in_gms'])
```

```
df_new.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10999 entries, 0 to 10998
Data columns (total 11 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Warehouse_block        10999 non-null  object
 1   Mode_of_Shipment       10999 non-null  object
 2   Customer_care_calls    10999 non-null  int64
 3   Customer_rating        10999 non-null  int64
 4   Prior_purchases        10999 non-null  int64
 5   Product_importance     10999 non-null  object
 6   Gender                 10999 non-null  object
 7   Reached_on_Time        10999 non-null  int64
 8   Cost_of_the_Product_tf 10999 non-null  float64
 9   Discount_offered_tf    10999 non-null  float64
 10  Weight_in_gms_tf       10999 non-null  float64
dtypes: float64(3), int64(4), object(4)
memory usage: 945.4+ KB
```

E. Feature encoding

Pada fitur ini, kolom-kolom diubah menggunakan metode Label encoding

```
# jenis_kelamin & pendidikan label encoding
mapping_jenis_kelamin = {
    'F' : 0,
    'M' : 1
}
```

```
mapping_warehouse_block = {
    'A' : 0,
    'B' : 1,
    'C' : 2,
    'D' : 3,
    'E' : 4
}
```

```
mapping_product_importance = {
    'low' : 0,
    'medium' : 1,
    'high' : 2
}
```

```
df_new['Gender'] = df_new['Gender'].map(mapping_jenis_kelamin)
df_new['Warehouse_block'] = df_new['Warehouse_block'].map(mapping_warehouse_block)
df_new['Product_importance'] = df_new['Product_importance'].map(mapping_product_importance)
```

E. Feature encoding

Pada fitur ini, kolom-kolom diubah menggunakan metode One Hot encoding

```
# one hot encoding
for cat in ['Mode_of_Shipment']:
    onehots = pd.get_dummies(df_new[cat], prefix=cat)
    df_new = df_new.join(onehots)
```

```
df_new.head(5)
```

	Warehouse_block	Mode_of_Shipment	Customer_care_calls	Customer_rating	Prior_purchases	Product_importance	Gender	Reached_on_Time	Cost_of_the_Product_tf	Discount_offered_tf	Weight_in_gms_tf	Mode_of_Shipment_Flight	Mode_of_Shipment_Road	Mode_of_Shipment_Ship
0	3	Flight	4	2	3	0	0	1	1.443699	1.269151	-1.468240	1	0	0
1	4	Flight	4	5	2	0	1	1	1.536534	0.150298	-0.333893	1	0	0
2	0	Flight	2	2	4	0	1	1	2.999592	-0.964595	-0.159002	1	0	0
3	1	Flight	3	3	4	1	1	1	-0.988708	0.488365	-1.502484	1	0	0
4	2	Flight	2	2	3	1	0	1	-0.259256	-0.803878	-0.703244	1	0	0

E. Feature encoding

```
df_new = df_new.drop(columns= 'Mode_of_Shipment')
```

```
df_new.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10999 entries, 0 to 10998
Data columns (total 13 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Warehouse_block         10999 non-null  int64  
 1   Customer_care_calls     10999 non-null  int64  
 2   Customer_rating         10999 non-null  int64  
 3   Prior_purchases         10999 non-null  int64  
 4   Product_importance      10999 non-null  int64  
 5   Gender                  10999 non-null  int64  
 6   Reached_on_Time         10999 non-null  int64  
 7   Cost_of_the_Product_tf  10999 non-null  float64
 8   Discount_offered_tf     10999 non-null  float64
 9   Weight_in_gms_tf        10999 non-null  float64
 10  Mode_of_Shipment_Flight 10999 non-null  uint8  
 11  Mode_of_Shipment_Road   10999 non-null  uint8  
 12  Mode_of_Shipment_Ship   10999 non-null  uint8  
dtypes: float64(3), int64(7), uint8(3)
memory usage: 891.6 KB
```

F. Handle class imbalance

```
print(df_new['Reached_on_Time'].value_counts(normalize=True))
```

```
1    0.596691
0    0.403309
Name: Reached_on_Time, dtype: float64
```

Tidak ada data yang harus dilakukan metode class imbalance dikarenakan target datanya masih balance

2. Feature Engineering

Pada fitur ini, kita melihat data-data yang ada pada kolom kita kemudian melakukan modifikasi terhadap fitur yang sudah ada, sehingga menjadi fitur baru yang nantinya dapat digunakan dalam proses pemodelan

A. Feature selection

- Membuang kolom id karena kolom id bersifat unique

B. Feature extraction

- (Logistic performance) Membuat pola berdasarkan weight in grams, dan memprediksi mode pengiriman yang cocok untuk berat yang sudah tertera pada data

2. Feature Engineering

C. Fitur Tambahan

- Tanggal order
- Tanggal delivery
- Lokasi order
- Tipe barang yang dikirim
- Jenis layanan