In [2]:
```
import numpy as np
```

In [3]:
```
import pandas as pd
```

In [4]:
```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('whitegrid')
%matplotlib inline
```

In [5]:
```
df = pd.read_csv('Train.csv')
```

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10999 entries, 0 to 10998
Data columns (total 12 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   ID                   10999 non-null  int64 
 1   Warehouse_block      10999 non-null  object
 2   Mode_of_Shipment     10999 non-null  object
 3   Customer_care_calls  10999 non-null  int64 
 4   Customer_rating      10999 non-null  int64 
 5   Cost_of_the_Product  10999 non-null  int64 
 6   Prior_purchases      10999 non-null  int64 
 7   Product_importance   10999 non-null  object
 8   Gender               10999 non-null  object
 9   Discount_offered     10999 non-null  int64 
 10  Weight_in_gms        10999 non-null  int64 
 11  Reached.on.Time_Y.N  10999 non-null  int64 
dtypes: int64(8), object(4)
memory usage: 1.0+ MB
```

In [7]:
```
df.head()
```

Out[7]:

	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

Number of orders for different warehouses

In [38]:
```
df["Warehouse_block"].value_counts()
```

Out[38]:
```
F    3666
D    1834
A    1833
B    1833
C    1833
Name: Warehouse_block, dtype: int64
```

Rating Count

In [9]:
```
df["Customer_rating"].value_counts()
```

Out[9]:
```
3    2239
1    2235
4    2189
5    2171
2    2165
Name: Customer_rating, dtype: int64
```

Numbers of on time/not on time order

In [10]:
```
df["Reached.on.Time_Y.N"].value_counts()
```

Out[10]:
```
1    6563
0    4436
Name: Reached.on.Time_Y.N, dtype: int64
```

Rating 5 with on time delivery

In [111]:
```
count = len(df[(df["Customer_rating"] == 5)&(df["Reached.on.Time_Y.N"] == 0)])
```

In [12]:
```
print(count)
```

```
854
```

Rating 1 with not on time delivery

In [13]:
```
count_1 = len(df[(df["Customer_rating"] == 1)&(df["Reached.on.Time_Y.N"] == 1)])
```

In [14]:
```
print(count_1)
```

```
1313
```

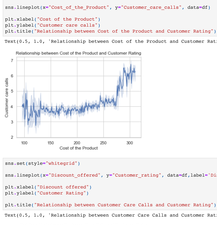
numbers of high importance and rating 5

In [15]:
```
count_importance = len(df[(df["Product_importance"] == "high")&(df["Customer_rating"] == 5)])
```

In [16]:
```
print(count_importance)
```

```
186
```

In [17]:
```
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(8, 6))

df['Mode_of_Shipment'].hist(ax=axes[0, 0])
axes[0, 0].set_title('Mode of Shipment')

df['Cost_of_the_Product'].hist(ax=axes[0, 1])
axes[0, 1].set_title('Cost of the Product')

df['Discount_offered'].hist(ax=axes[1, 0])
axes[1, 0].set_title('Discount offered')

df['Weight_in_gms'].hist(ax=axes[1, 1])
axes[1, 1].set_title('Weight in gms')

plt.tight_layout()
plt.show()
```



In [18]:
```
cont_table = pd.crosstab(df['Product_importance'], df['Customer_rating'])

print(cont_table)

plt.scatter(df['Product_importance'], df['Customer_rating'])
plt.title('Customer Ratings by Product Importance')
plt.xlabel('Product Importance')
plt.ylabel('Customer Rating')
plt.show()
```

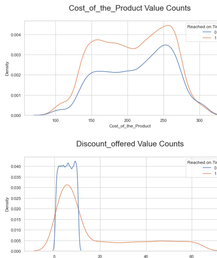
```
Customer_rating     1     2     3     4     5
Product_importance                           
high              186   199   184   193   186
low              1117  1315  1054  1060  1051
medium            932   951  1001   926   934
```

In [19]:
```
sns.set(style='whitegrid')

sns.lineplot(x='Customer_care_calls', y='Customer_rating', data=df, label='Customer care calls')

plt.xlabel('Customer Care Calls')
plt.ylabel('Customer Rating')

plt.title('Relationship between Customer Care Calls and Customer Rating')
```

Out[19]:
```
Text(0.5, 1.0, 'Relationship between Customer Care Calls and Customer Rating')
```

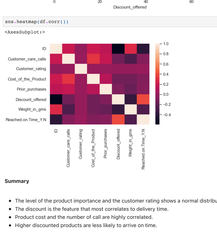
In [20]:
```
sns.set(style='whitegrid')

sns.lineplot(x='Cost_of_the_Product', y='Customer_care_calls', data=df)

plt.xlabel('Cost of the Product')
plt.ylabel('Customer care calls')
plt.title('Relationship between Cost of the Product and Customer Rating')
```

Out[20]:
```
Text(0.5, 1.0, 'Relationship between Cost of the Product and Customer Rating')
```



In [21]:
```
sns.set(style='whitegrid')

sns.lineplot(x='Discount_offered', y='Customer_rating', data=df,label='Discount offered')

plt.xlabel('Discount offered')
plt.ylabel('Customer Rating')

plt.title('Relationship between Customer Care Calls and Customer Rating')
```

Out[21]:
```
Text(0.5, 1.0, 'Relationship between Customer Care Calls and Customer Rating')
```



In [26]:
```
sns.set(style='whitegrid')

sns.barplot(x='Product_importance', y='Reached.on.Time_Y.N', data=df)

plt.xlabel('Product_importance')
plt.ylabel('Reached.on.Time_Y.N')

plt.title('Relationship between Product Importance and Delivery Time')
```

Out[26]:
```
Text(0.5, 1.0, 'Relationship between Product Importance and Delivery Time')
```

relationships between different factors and delivery time

In [28]:
```
cols = [ 'Customer_care_calls', 'Customer_rating','Cost_of_the_Product','Prior_purchases','Discount_offered','Weight_in_gms']

plt.figure(figsize = (18, 20))
plotnumber = 1

for i in range(len(cols)):
    if plotnumber <= 6:
        ax = plt.subplot(3, 2, plotnumber)
        sns.kdeplot(df[cols[i]], data = df, ax = ax,hue='Reached.on.Time_Y.N')
        plt.title("'"+cols[i]+' Value Counts\n', fontsize = 21)
    plotnumber += 1
plt.tight_layout()
plt.show()
```


In [29]:
```
sns.heatmap(df.corr())
```

Out[29]:
```
<AxesSubplot:>
```


Summary

- The level of the product importance and the customer rating shows a normal distribution.
- The discount is the feature that most correlates to delivery time.
- Product cost and the number of call are highly correlated.
- Higher discounted products are less likely to arrive on time.

Suggestion

Nearly 60% of the product were not delivered on time, which means the company should adjust the delivery strategy. Warehouse F handle 2 times more orders than others, that could be more reasonable to distribute the orders more normal. Despite the number of calls from customers seems normal, the company should reduce the number to offer a better service and keep more customers to make purchases constantly.If there is more data provided, I would like to analyze the reason why the products with discount are more likely to deliver on time, which my best surmise would be the products with discounts occupied a majority of the sales which led the entire sales figures.