CAPSTONE PROJECT

PREDICT THE RISK OF LATE DELIVERY FOR CUSTOMER ORDERS USING MACHINE LEARNING TECHNIQUES

FINAL REPORT

Mentored by :

DR. DEBASISH ROY

Submitted by: (Group – 5)

Mr. BRANESH KUMAR
Mr. HARSHAL MANSUKLAL JAIN
Mr. HARIHARAN
Mr. VIJAY VENKATRAJ
Mr. ARAVIND UNNIKRISHNAN

Table of Contents:

Overview:

Industry Background and Key Objectives

Late delivery risk prediction using machine learning is a technique that involves building a predictive model to identify the probability of an order being delivered late based on historical order data. This model can be trained using various features such as product type, shipping duration, and customer location. By analyzing past orders, the model can learn to accurately predict the likelihood of a late delivery and help businesses to take preventive measures to avoid late deliveries.

Business problem statement:

1) Business Problem Understanding

The business problem of late delivery risk prediction using machine learning is to identify orders that are at high risk of being delivered late to avoid potential customer dissatisfaction and revenue loss. By predicting late delivery risk, businesses can take preventive measures such as reallocating resources, optimizing delivery routes, and proactively managing customer expectations to ensure timely deliveries. This can improve customer satisfaction and retention while reducing the costs associated with late deliveries, such as shipping fees, order cancellations, and negative customer reviews.

Main challenges involved in Late Delivery Prediction are:

- Lack of visibility: Businesses may not have complete visibility into the supply chain, leading to delays and disruptions that can affect delivery times.
- Inaccurate demand forecasting: If businesses do not have an accurate forecast of demand, they may not have enough resources or inventory to fulfill orders on time.
- Traffic and weather conditions: External factors such as traffic congestion and severe weather can delay deliveries and make it difficult to meet customer expectations.

- Capacity constraints: If businesses do not have enough resources or capacity to fulfill orders, they may not be able to meet delivery deadlines.
- Poor communication: Inadequate communication between different departments or stakeholders involved in the delivery process can lead to delays and errors.
- Inefficient processes: Inefficient delivery processes such as manual data entry or inefficient routing can cause delays and increase the risk of late deliveries.
- Unexpected events: nexpected events such as equipment failure or employee absence can disrupt the delivery process and lead to late deliveries.

2) Business Objective

Broadly speaking, recent developments in machine learning techniques and data mining has led to an interest of implementing these techniques in various fields. The e-commerce space is no different in this regard. Potentially, the implementation of machine learning techniques could lead greater overall profitability in the business.

It is important for E-Commerce platforms to better understand the trends and key reasons involved in customers subsequently rejecting orders they have placed. This will not only help the business identify potential pain

points in the shopping experience, but might also help in finding ways in which customer returns can be reduced, and these associated costs can be reduced. Ideally, both a reduction in order rejection volumes and an improvement in customer shopping experience can be achieved by implementing suggestions derived from such an analysis.

Data Dictionary:

Fields	Description
Type	Type of transaction made
Days for shipping (real)	Actual shipping days of the purchased product
Days for shipment (scheduled)	Days of scheduled delivery of the purchased product
Benefit per order	Earnings per order placed
Sales per customer	Total sales per customer made per customer
Delivery Status	Delivery status of orders: Advance shipping , Late delivery , Shipping canceled , Shipping on time
Late_delivery_risk	Categorical variable that indicates if sending is late (1), it is not late (0).
Category Id	Product category code
Category Name	Description of the product category
Customer City	City where the customer made the purchase
Customer Country	Country where the customer made the purchase
Customer Email	Customer's email
Customer Fname	Customer name
Customer Id	Customer ID
Customer Lname	Customer last name
Customer Password	Masked customer key
Customer Segment	Types of Customers: Consumer, Corporate, Home Office
Customer State	State to which the store where the purchase is registered belongs
Customer Street	Street to which the store where the purchase is registered belongs
Customer Zipcode	Customer Zipcode
Department Id	Department code of store
Department Name	Department name of store
Latitude	Latitude corresponding to location of store
Longitude	Longitude corresponding to location of store
Market	Market to where the order is delivered: Africa, Europe, LATAM, Pacific Asia, USCA
Order City	Destination city of the order
Order Country	Destination country of the order
Order Customer Id	Customer order code
order date (Date Orders)	Date on which the order is made
Order Id	Order code
Order Item Card prod Id	Product code generated through the RFID reader
Order Item Discount	Order item discount value
Order Item Discount Rate	Order item discount percentage
Order Item Id	Order item code
Order Item Product Price	Price of products without discount
Order Item Profit Ratio	Order Item Profit Ratio
Order Item Quantity	Number of products per order
Sales	Value in sales

Order Item Total	Total amount per order
Order Profit Per Order	Order Profit Per Order
Order Region	Region of the world where the order is delivered: Southeast Asia, South Asia, Oceania, Eastern Asia, West Asia, West of USA, US Center, West Africa, Central Africa, North Africa, Western Europe, Northern, Caribbean, South America, East Africa, Southern Europe, East of USA, Canada, Southern Africa, Central Asia, Europe, Central America, Eastern Europe, South of USA
Order State	State of the region where the order is delivered
Order Status	Order Status : Complete , Pending , Closed , Pending payment ,Cancelled, Processing ,Suspected fraud, On hold, Payment review
Product Card Id	Product code
Product Category Id	Product category code
Product Description	Product Description
Product Image	Link of visit and purchase of the product
Product Name	Product Name
Product Price	Product Price
Product Status	Status of the product stock: If it is 1 not available, 0 the product is available
Shipping date (Date Orders)	Exact date and time of shipment
Shipping Mode	The following shipping modes are presented: Standard Class, First Class, Second Class, Same Day

4.2 Shape of The Dataset:

```
df.shape
```

```
(180519, 53)
```

4.3 Variable Categorization:

Variables	Count
Numerical Variables	28
Categorical Variables	24
Target Variables (Binary- 0 and 1)	1
Total Variables (Columns)	53

Data Pre-processing:

We begin by reading in the default csv file and getting a sense of the overall size and features we will be working with.

Our default data file includes:

Number of Columns: 53
Total Number of Records: 180519

Dealing With Missing Values:

Our initial search helps us identify missing values. The columns ‘ Product Description’ and ‘Order Zipcode’ have data missing greater than 86%.

```
df_supply_chain.isnull().sum()[df_supply_chain.isnull().sum()*100/len(df_supply_chain) > 10]
```

```
Order Zipcode          155679
Product Description    180519
dtype: int64
```

	Total	Percentage of Missing Values
Product Description	180519	100.0000
Order Zipcode	155679	86.2397
Customer Lname	8	0.0044
Customer Zipcode	3	0.0017

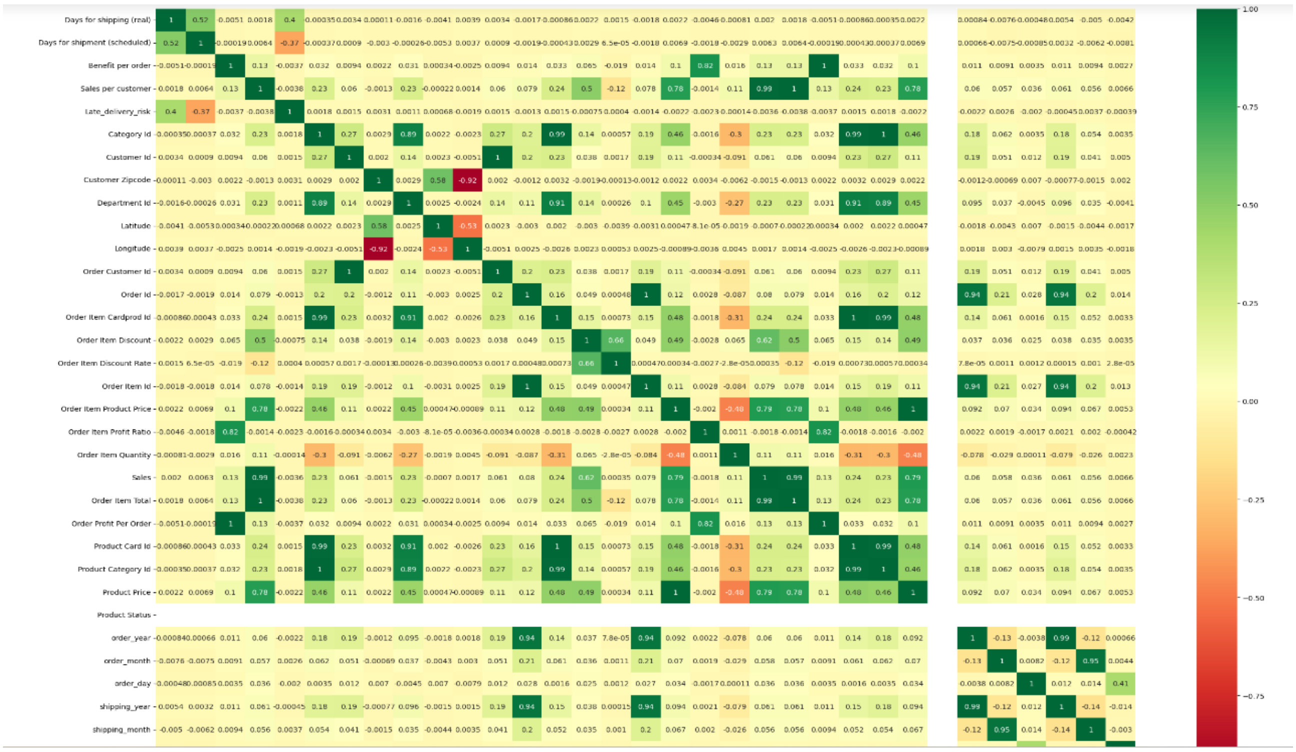
Dropping the columns where the null values are more than 86%.
So, we will drop the columns, ‘Product Description’ and ‘Order Zipcode’.
However, Product Description columns is not important for model prediction.

Converting the object datatype to datetime and Extracting Year, Month and Day:

Converting data, Extracting Year, Month, Day from Date and updating column name.

```
In [8]: # from the that we are extracting the date only
        df_supply_chain['order_date'] = pd.to_datetime(df_supply_chain['order date (DateOrders)']).dt.date
        df_supply_chain['shipping_date']=pd.to_datetime(df_supply_chain['shipping date (DateOrders)']).dt.date


        # Converting categorical features that represent date and time to datetime datatype.
        df_supply_chain['order_date'] = pd.to_datetime(df_supply_chain['order_date'])
        df_supply_chain['shipping_date']=pd.to_datetime(df_supply_chain['shipping_date'])



        # Handling Time and date variables

        df_supply_chain['order_year'] = pd.DatetimeIndex(df_supply_chain['order_date']).year
        df_supply_chain['order_month'] = pd.DatetimeIndex(df_supply_chain['order_date']).month
        df_supply_chain['order_day'] = pd.DatetimeIndex(df_supply_chain['order_date']).day
        df_supply_chain['shipping_year'] = pd.DatetimeIndex(df_supply_chain['shipping_date']).year
        df_supply_chain['shipping_month'] = pd.DatetimeIndex(df_supply_chain['shipping_date']).month
        df_supply_chain['shipping_day'] = pd.DatetimeIndex(df_supply_chain['shipping_date']).day


        # Droping the old column  and adding new column instead of the shipping date (DateOrders) to shipping_date ,
        # order date (DateOrders) instead of order_date
        df_supply_chain=df_supply_chain.drop(['shipping date (DateOrders)','order date (DateOrders)'],axis=1)

In [9]: df_supply_chain=df_supply_chain.drop(['Order Zipcode','Product Description'],axis=1)
```

Checking multicollinearity between features to check for duplicate features:



After comparing the values in columns pairs having co-relation of 1, we can see that all values are equal for these pair of columns. We can drop one of the columns from each of the above pairs as they are duplicate columns of each other

```
1. Customer Id                 1.0
   Order Customer Id           1.0

2. Sales per customer          1.0
   Order Item Total            1.0

3. Benefit per order           1.0
   Order Profit Per Order      1.0

4. Order Item Cardprod Id      1.0
   Product Card Id             1.0

5. Category Id                 1.0
   Product Category Id         1.0

6. Order Item Product Price    1.0
   Product Price               1.0
```

Dropping of columns:

FEATURES DROPPED	REASON FOR DROPPING
Product Description	It contains 100% null values. This column did not provide any useful information for analysis and was therefore deemed irrelevant.
Order Zipcode	It contains 86.23% null values. This column did not provide any useful information for analysis and was therefore deemed irrelevant.
Customer Email, Customer Password, Customer Fname & Customer Lname	They were not expected to have a significant impact on predicting the target variable 'late delivery risk'. These columns were deemed irrelevant to the analysis.
Benefit per order	It is giving the same information as "order profit per order".
Category Id and Category Name	It is giving the same information about "Product Category ID".
Order Customer ID	It is giving the same information as "Customer Id"
Sales per customer	It is giving the same information as "Order Item Total"
Department Name	It is giving the same information as "Department ID"
Customer Street	The information it contained was redundant with other columns such as 'Customer Zip code', 'Customer City', 'Customer State', and 'Customer Country'.

Product Images	They do not provide any direct information related to the delivery process
Order date and Shipping date	They contained redundant information with the columns'Days for shipping(real)' and'Days for shipment(scheduled)'. The columns 'Days for shipping (real)' and 'Days for shipment (scheduled)' for analysis were retained as they provided more relevant information about the shipping and delivery process.
Product Status	All values in "Product Status" are zero so given feature is not giving any information.
Order Id, Order Item Id, & Product Card Id	They were not expected to have a significant impact on predicting the target variable 'late delivery risk'. These columns might have been redundant or irrelevant to the analysis and could also have had data quality issues.

Also, we can drop Customer city, Longitude and Latitude as they are not needed in building our model.

Finding the Delivery Time Difference:

```
In [33]: # Calculate the delivery time difference
delivery_time_difference = df_supply_chain["Days for shipping (real)"] - df_supply_chain["Days for shipment (scheduled)"]

# Insert the new column at the desired position
df_supply_chain.insert(3, "Delivery_Time_Difference", delivery_time_difference)
```

```
In [34]: df_supply_chain['Delivery_Time_Difference'].unique()

Out[34]: array([-1,  1,  0, -2,  2,  4,  3], dtype=int64)
```

Summary of Data:

```
: df_supply_chain.describe()
```

	Days for shipping (real)	Days for shipment (scheduled)	Delivery_Time_Difference	Sales per customer	Late_delivery_risk	Order Item Discount	Order Item Quantity	Order Profit Per Order	Product Price
count	180519.000000	180519.000000	180519.000000	180519.000000	180519.000000	180519.000000	180519.000000	180519.000000	180519.000000
mean	3.497654	2.931847	0.565807	183.107609	0.548291	20.664741	2.127638	21.974989	141.232550
std	1.623722	1.374449	1.490966	120.043670	0.497664	21.800901	1.453451	104.433526	139.732492
min	0.000000	0.000000	-2.000000	7.490000	0.000000	0.000000	1.000000	-4274.979980	9.990000
25%	2.000000	2.000000	0.000000	104.379997	0.000000	5.400000	1.000000	7.000000	50.000000
50%	3.000000	4.000000	1.000000	163.990005	1.000000	14.000000	1.000000	31.520000	59.990002
75%	5.000000	4.000000	1.000000	247.399994	1.000000	29.990000	3.000000	64.800003	199.990005
max	6.000000	4.000000	4.000000	1939.989990	1.000000	500.000000	5.000000	911.799988	1999.989990

1. In real, maximum days required to shipping is 6 but as per the schedule is 4. From the data, 25% of delivery estimation is on a time (day of scheduled) some order they delivered on same day of order.

2. From Delivery time difference column, some order is delivered in 2 days before the delivery date and some orders are delivered 4 days late from the scheduled time.

3. Average Sale per customer is 183.10-dollar, minimum sale is 7.5 dollar and maximum is 1940 dollar. 50% of customers spend 164 dollars.

4. 54.8% item delivery is late and only 45.2 % items are delivered on time or schedule time.

5. Highest discount given by sale is 500 dollar and average discount per customer is 20.66 dollar. 25% of people get 5.4 dollar as discount this value dramatically change for every quartile for 50% value stand at 14, for 75% the discount was 29.9 dollar.

6. Most of the customers are interested to buy only one product and some customers are buying more than 1, the highest quantity of buying items is 5.

7. Highest benefit per order is 911 dollar and loss are 4275-dollar, average benefit per order is 22 dollars.

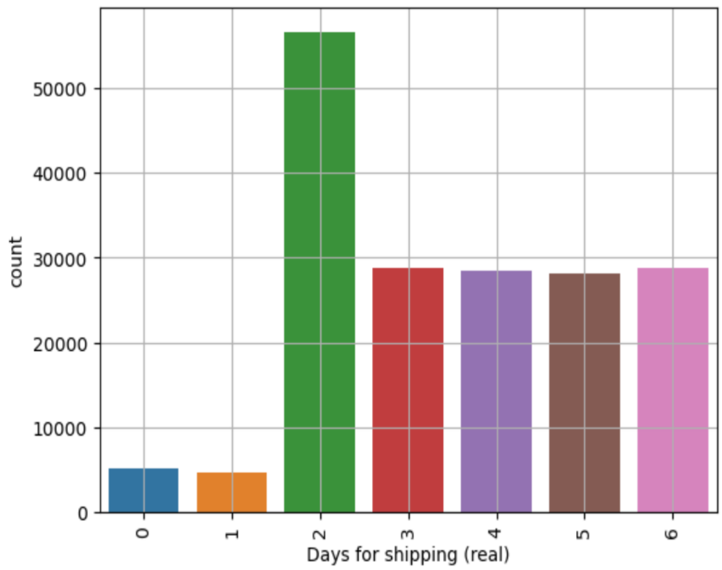
8. Maximum price of product is 2000 dollar and minimum price of product is 10-dollar, average price per product is 141 dollars approximately.

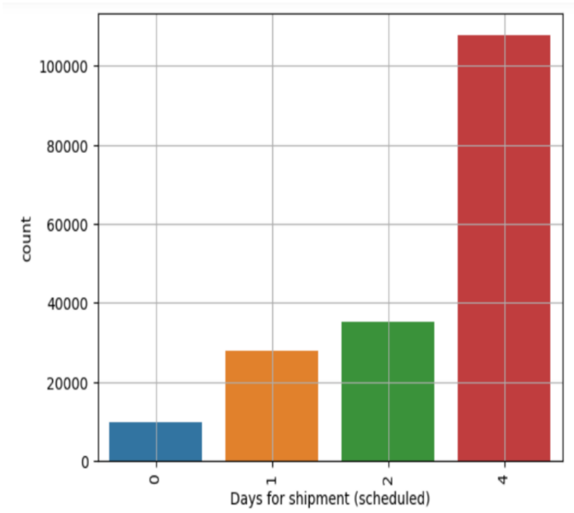
Initial Data Exploration - analyzing Relationships between the Variables:

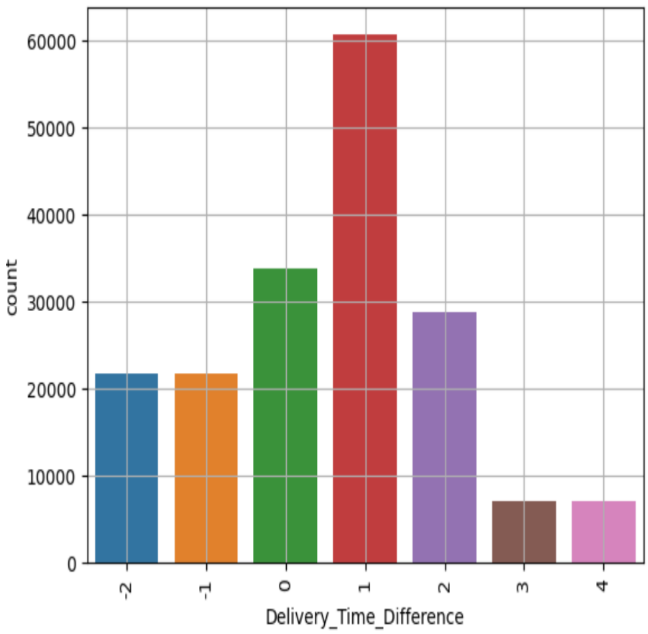
Univariate analysis:

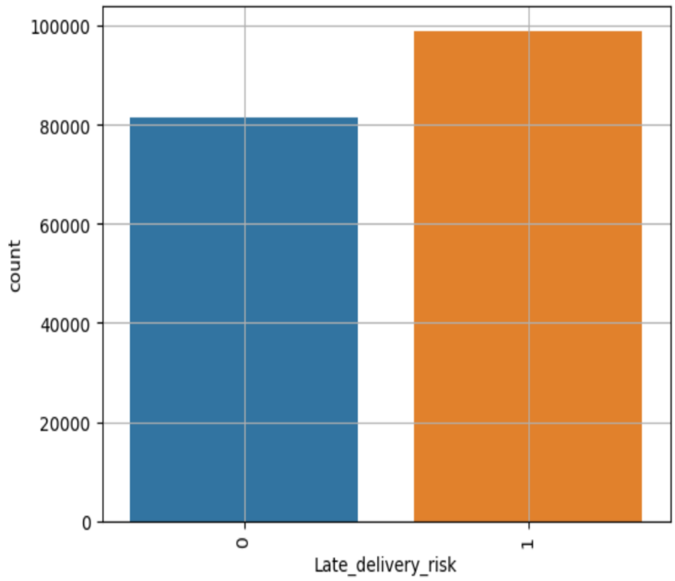
Let us begin by charting the updated categorical columns

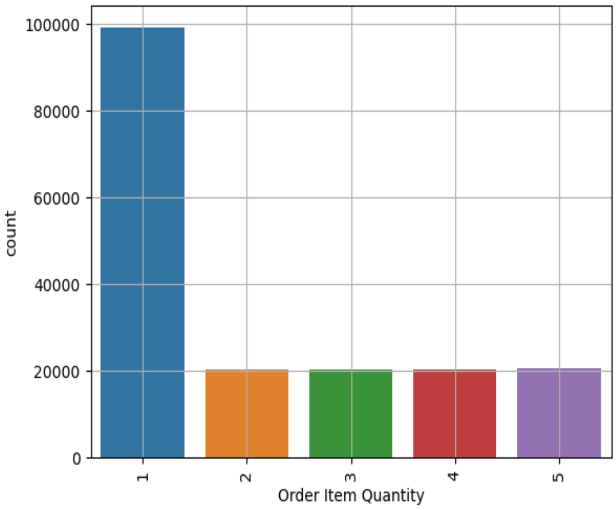
Below, we can see that a majority of the orders at getting delivered late. Comparatively advance shipping orders are also getting delivered late.

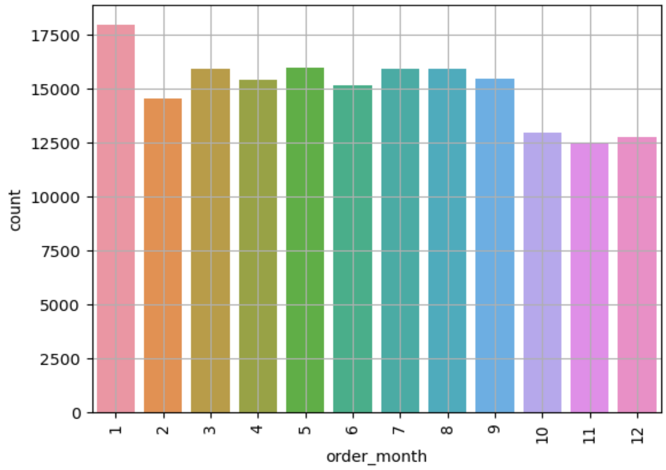


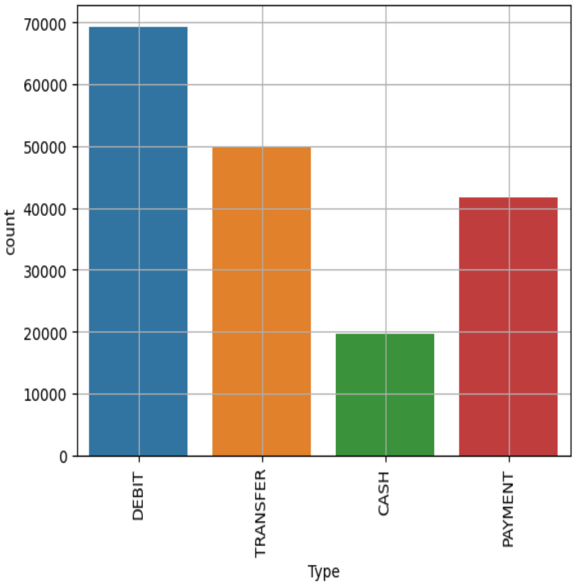


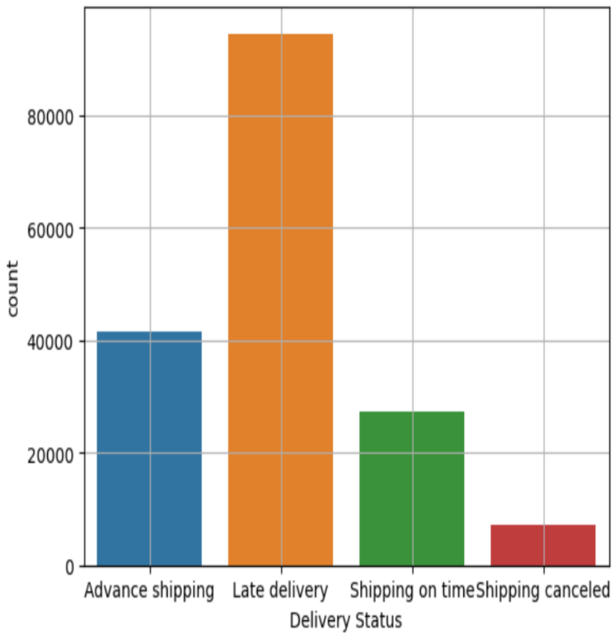


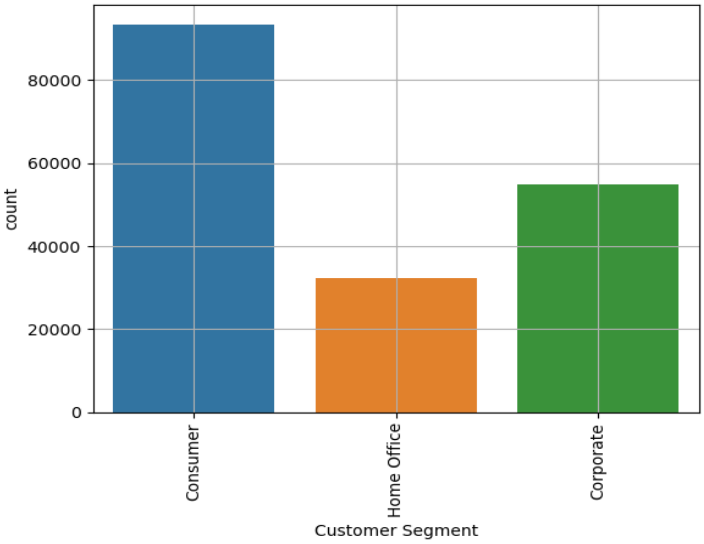


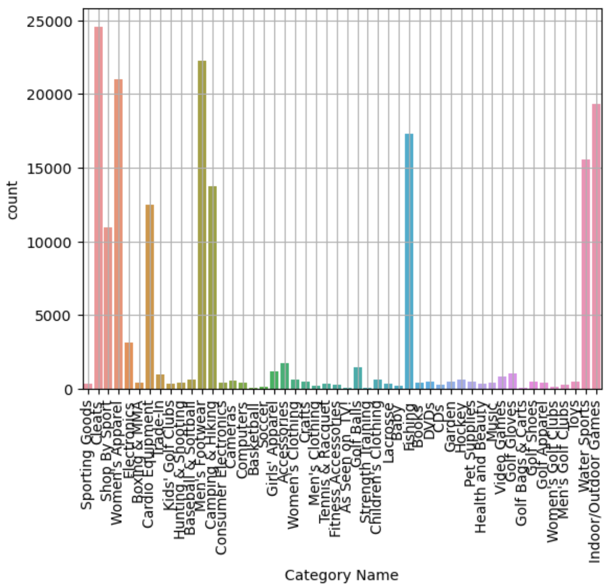

count
100000
80000
60000
40000
20000
0
1
2
3
4
5
Order Item Quantity

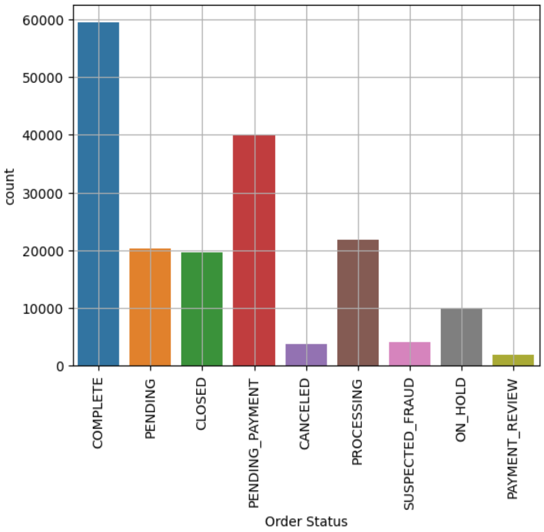

count
17500
15000
12500
10000
7500
5000
2500
0
1
2
3
4
5
6
7
8
9
10
11
12
order_month

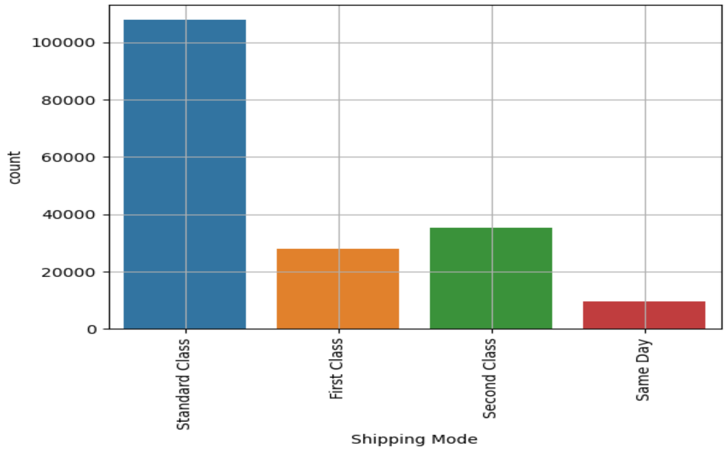

count
70000
60000
50000
40000
30000
20000
10000
0
DEBIT
TRANSFER
CASH
PAYMENT
Type

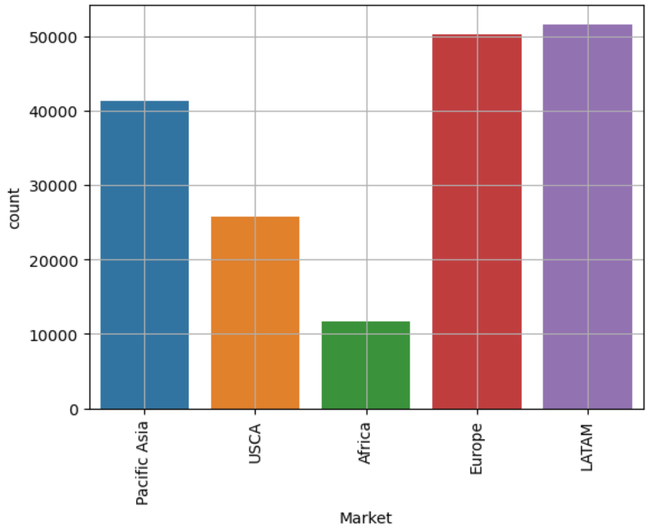

count
80000
60000
40000
20000
0
Advance shipping
Late delivery
Shipping on time
Shipping canceled
Delivery Status

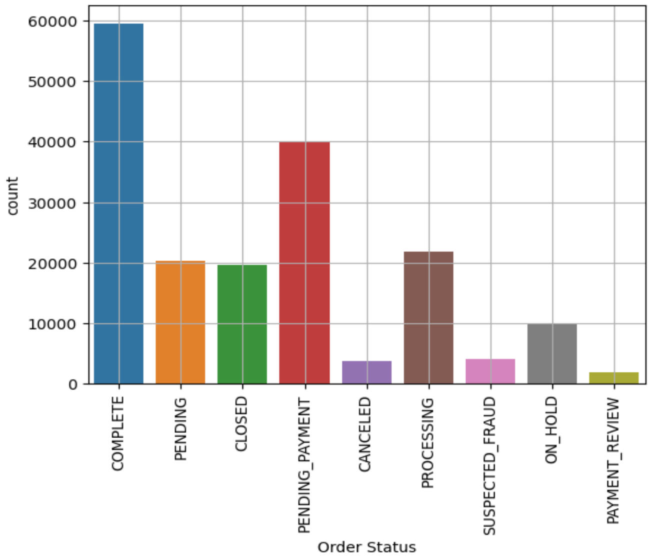

count
80000
60000
40000
20000
0
Consumer
Home Office
Corporate
Customer Segment


count
25000
20000
15000
10000
5000
0
Sporting Goods
Cleats
Shop By Sport
Women's Apparel
Electronics
Boxing & MMA
Cardio Equipment
Trade-In
Kids' Golf Clubs
Hunting & Shooting
Baseball & Softball
Men's Footwear
Camping & Hiking
Consumer Electronics
Cameras
Computers
Basketball
Soccer
Girls' Apparel
Accessories
Women's Clothing
Crafts
Men's Clothing
Tennis & Racquet
Fitness Accessories
As Seen on TV!
Golf Balls
Strength Training
Children's Clothing
Lacrosse
Baby
Fishing
Books
DVDs
CDs
Garden
Hockey
Pet Supplies
Health and Beauty
Music
Video Games
Golf Gloves
Golf Bags & Carts
Golf Shoes
Golf Apparel
Women's Golf Clubs
Men's Golf Clubs
Toys
Water Sports
Indoor/Outdoor Games
Category Name

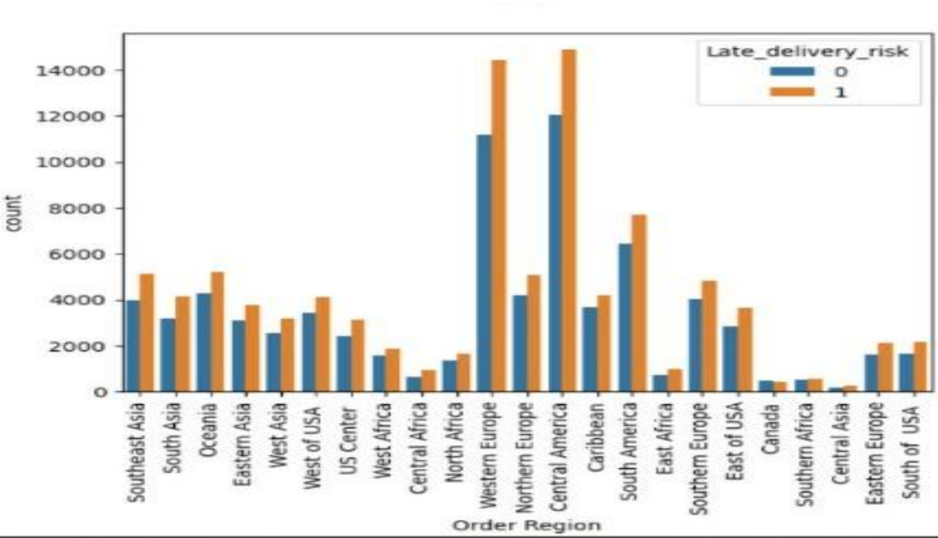








Observations



1. More than 55000 delivery day of shipping (real) is 2 days. day 3,4,5,6 are normally equally distributed in between 28000 to 29000.
2. According to schedule all order should be delivered within 4 days. Delivery within 4 days, within 2 days, within 1 days, and same day delivery count around 105000,48000,25000,5000 respectively.
3. Approx. 61000 deliveries late by 1 days and 45000 delivery late between 2 and 4 days. before the schedule item delivered is 45000 approximately.
4. From Late Delivery Columns, we can infer that Late Delivery Count stand at 98000 approx. and Delivery on Time count above 80000
5. 55% (98000 approx.) Customer preferred to buy 1 quantity and in 2,3,4,5 quantity buyer customer are equally distributed in remaining 45%
6. From order year attribute we can say the sale is decrease between 2016 and 2018
7. Most of customer preferred to by product using Debit Card (approx. 70000 people). only 20000 people choose payment method as a cash, 50000 customer use Bank Transfer method remaining use another payment method as a payment.
8. Cleats was the top selling category and Golf Bags & Carts falls in the lowest selling category
9. The Most customer from Consumer (around 88000) then followed by corporate stand at 57000 and 37000 approx. from home office category.
10. Central America orders highest and second most order region is West Europe.
11. More than 100000 customer from LATAM and Europe market.
12. The least order region are Canada, Central Asia, Southern Africa, West Africa, East Africa and Central Africa.
13. Due to the payment pending, pending, processing the order is still not complete.
14. There is 4 type of shipping mode Standard Class, Second class, First class, same day - 112000,36000,23000,7000 approx. respectively.

Finally, looking at the order status and comparing by region, we recognize that a majority of the orders are from Europe, Latam locations and most had been shipped at the point of data collection.

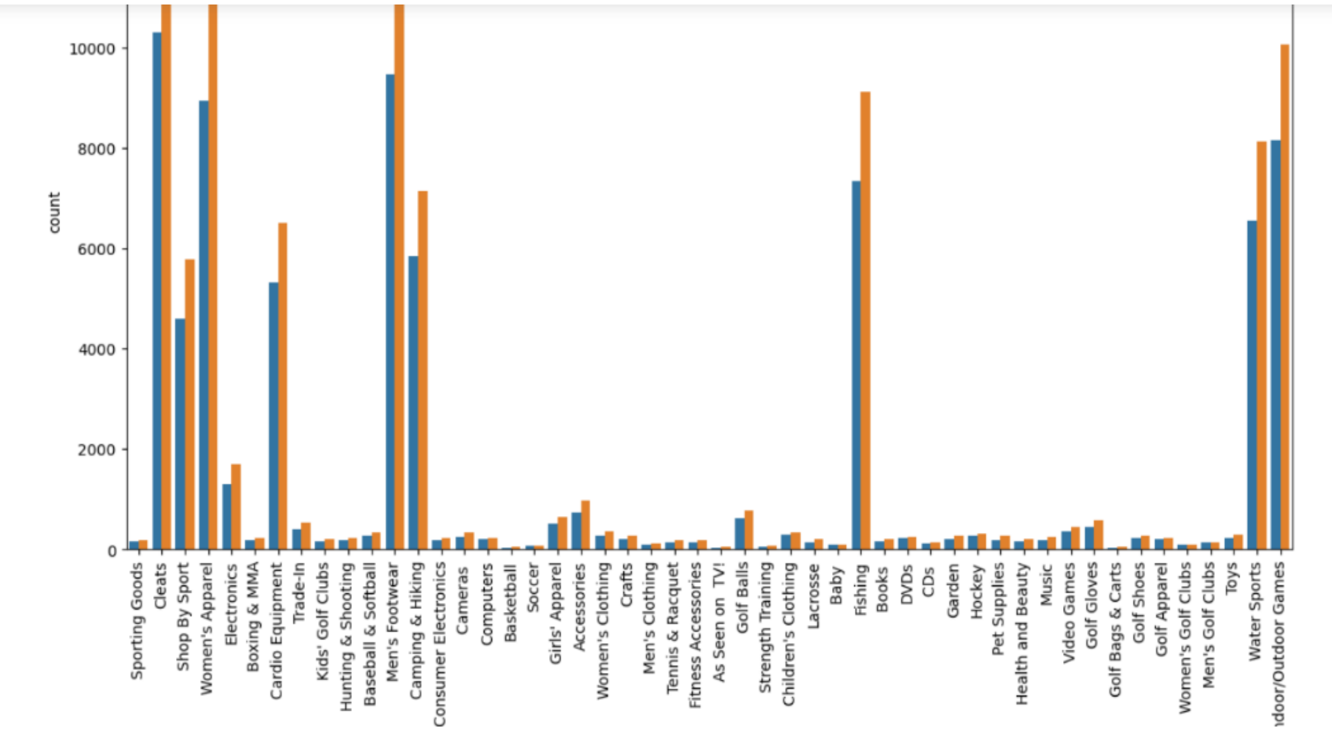
Displaying the target variable:

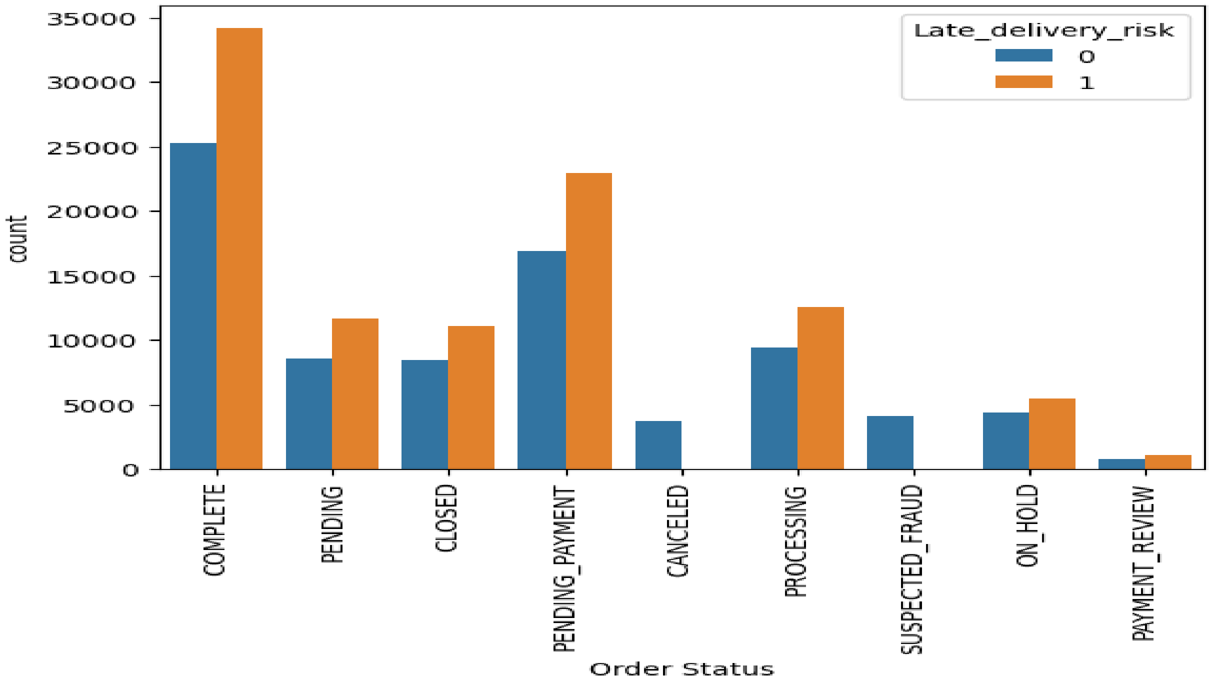
We reconfirm the fact that most orders are categorized under the Late delivery risk section, with about 54.8% of the orders being categorized as ‘Late delivered’. We hopeto see a reasonable reduction in the % share for ‘Late delivered’ post implementation of suggestions made via this analysis.

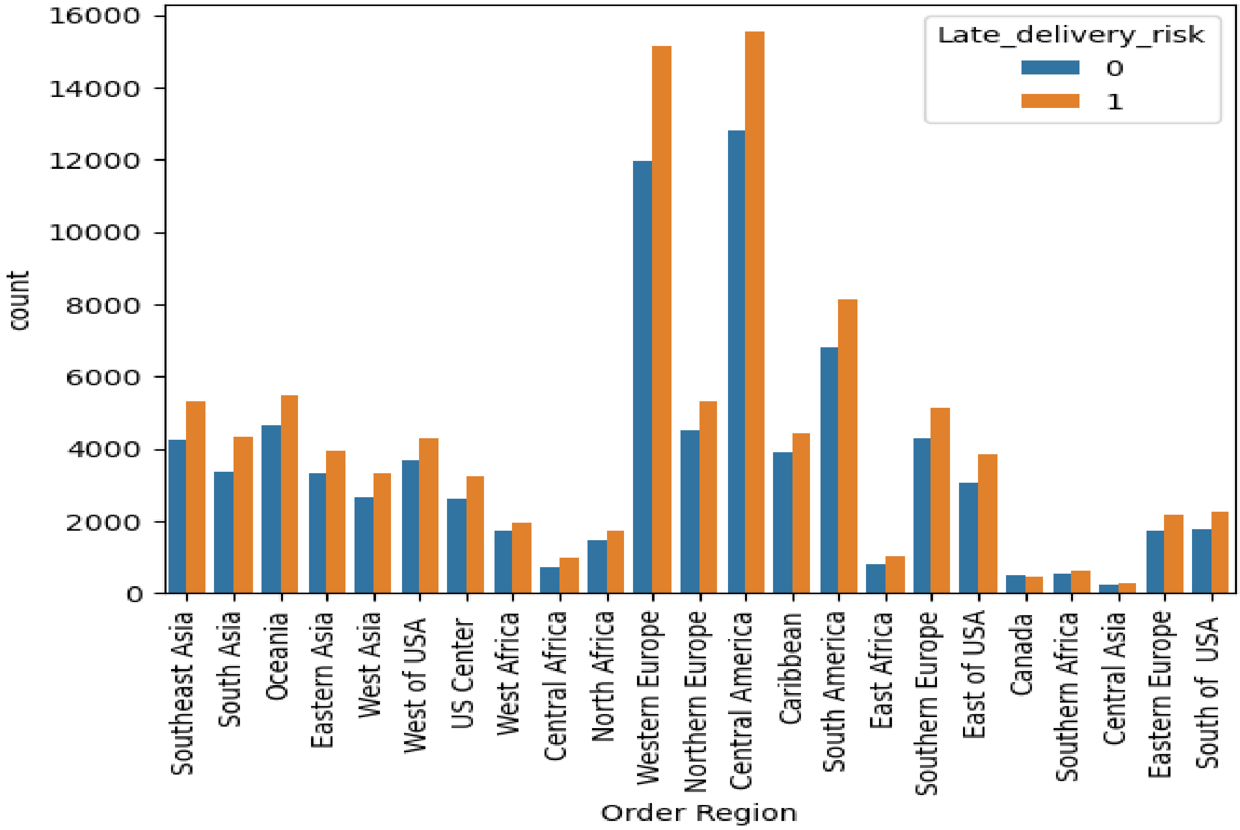
Bivariate analysis:

```
In [14]:  1  df_supply_chain_1['Late_delivery_risk'].value_counts(normalize=True)*100

Out[14]: 1    54.829132
         0    45.170868
         Name: Late_delivery_risk, dtype: float64
```

Here we will attempt to understand how the available columns are impacting each other on a one-to-one basis.







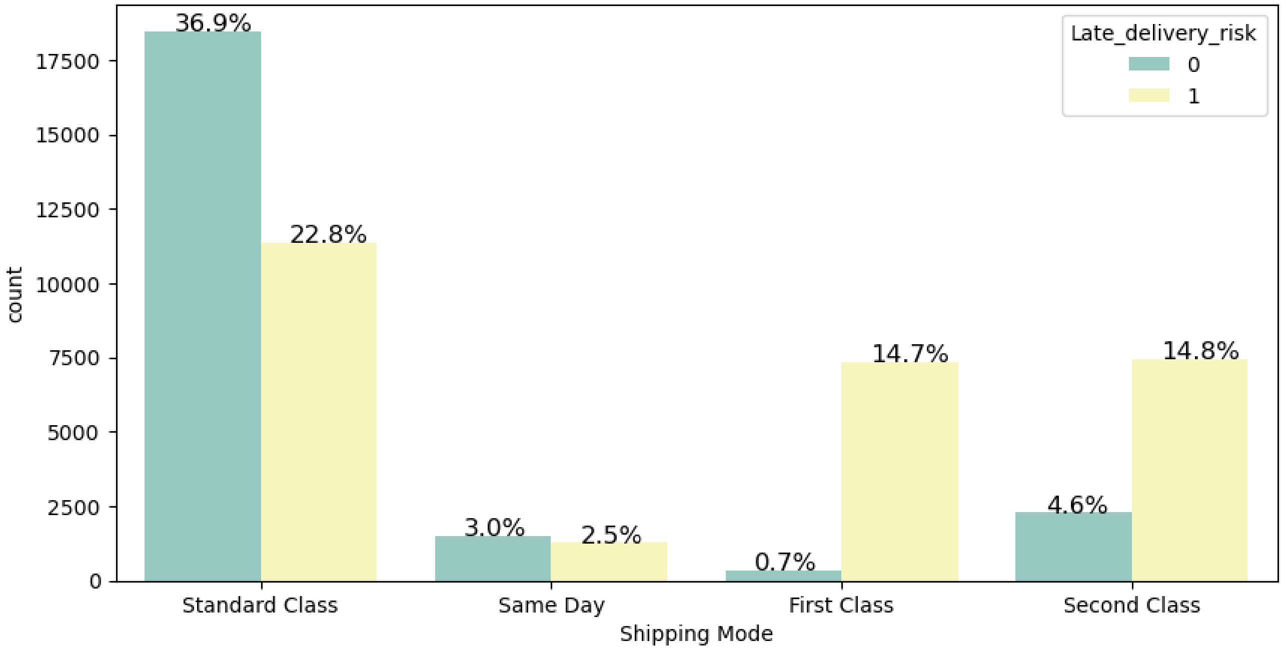
<u>Now, let us look at how categorical column data is distributed</u>

From graph, every Product Category has maximum count goes to Late Delivery as compared to On Delivered. Order Status is also same as previous thoughts like the count is maximum in each segment except Order Canceled and fraud suspected.

In order status also same as previou0073 thoughts like the count is maximum in each segment except Order Cancelled and fraud suspected

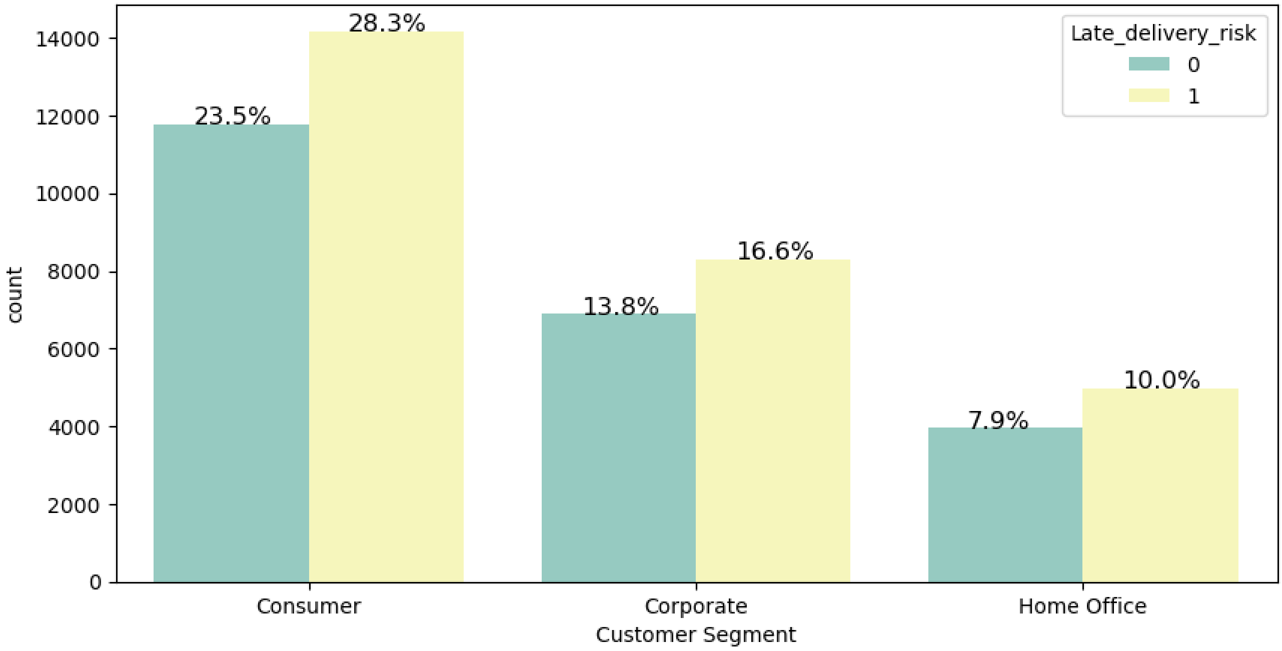
If we consider the Region, the Western Europe and Central America is higher order placed than another region, count is 12000 and 13000 respectively. late delivery stand at 15000 and 15500 respectively.

Shipping Mode vs Late delivery risk:



Standard Class is the most popular shipping mode. Late delivery is even observed in first class shipping mode.

Market vs Late delivery risk:



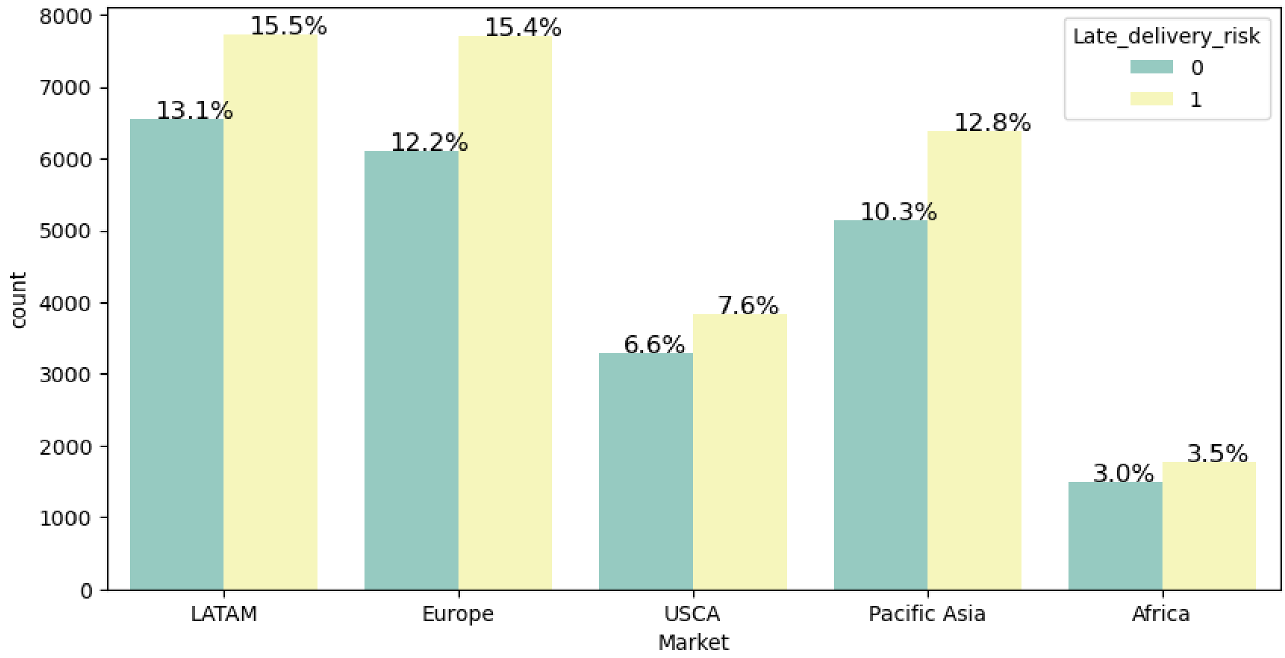
OBSERVATION:

- **28.3%** late delivery to Consumer and **23.5%** are not late delivery to Consumer

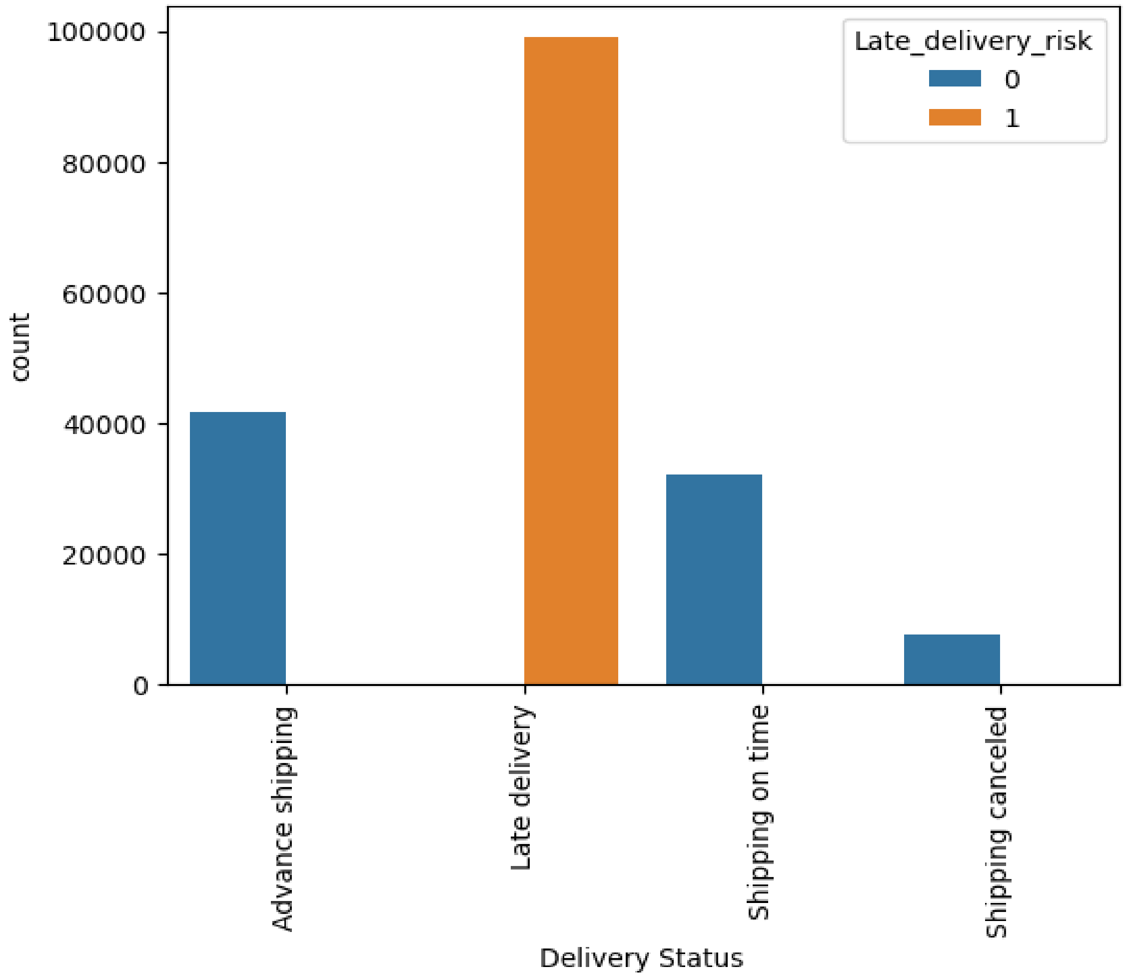
- **16.6%** late delivery to Corporate Customer Segment and **13.8%** are not late delivery to Corporate Customer Segment.
- **10.0%** late delivery to Home Office and **7.9** are not late delivery to Home Office

greatlearning
Learning for Life

Market vs Late delivery risk:



- Same number of percentage **(approx. 15.5%)** of late delivery from EUROPE and LATAM
- **3.5%** of late delivery from Africa because anyhow less number of delivery are coming from Africa.



- 54.8% orders were delivered late.

]

- 23% orders were shipped in advance.
- 17.8% orders were shipped on time.
- 4.3% shipping were cancelled.

Shipping Type Distribution:

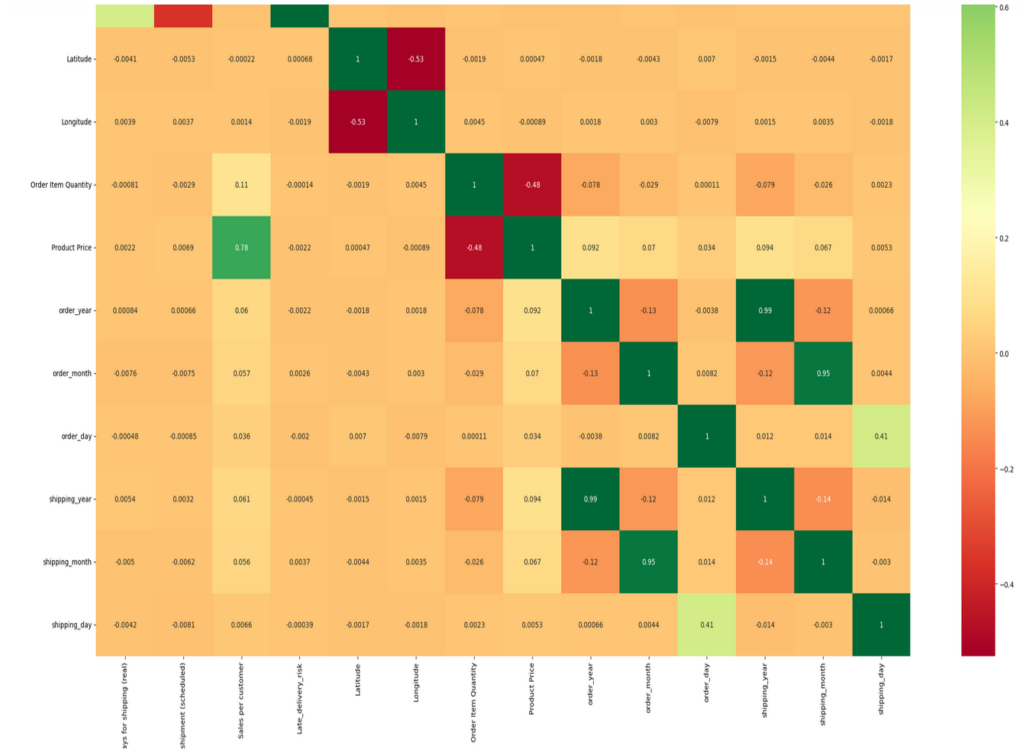
- 51.8% of customers are consumers.
- 30.4% are corporates.
- 17.9% people are from home, office category.

Multivariate Analysis:

In this section we aim to visualize the relationship between our target variable and features more broadly.

For instance, it might be useful to check how order rejection is affected when distributed by clothing-type/category and the average amount per order for the given clothing categories.

Similar to above, we visualized order rejection proportions vs the various order sizes that have been selected by customers.



Outlier Treatment:

While we have worked to treat missing values in our dataset, we have taken a different approach when it comes to outliers, given the type of data available and the scope of analysis that we aim to undertake.

For example, when it comes to extreme values in column, it makesmore sense to just leave them as it is, rather than trying to filter them out or cappingthe values. Later on we analyze rejections vs good perspective and make an inference accordingly.

Treatment of Imbalanced Data

Common consequences suggests that a proportion of at least 30:70 should be presenting a binary target variable.

Similar to our decision on outliers, we have made certain considerations for our target variable, and not attempted to treat imbalanced immediately at the outset.

We have decided to use the data in its original proportion for our base model, and based on its performance will make decisions on whether and how to adjust the imbalance at a later stage.

Statistical Analysis:

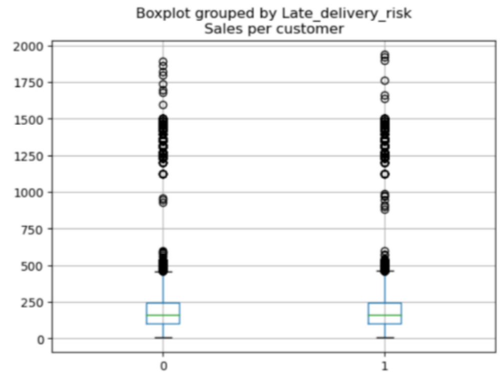
```
from scipy import stats
from scipy.stats import chi2_contingency,ttest_ind
cat=[]
num=[]
for i in df_supply_chain:
    if df_supply_chain[i].dtypes=='object':
        observed=pd.crosstab(df_supply_chain[i],df_supply_chain['Late_delivery_risk'])
        z_stat,p_val,ddof,exp=chi2_contingency(observed,correction=False)
        print(f'p value for {i} and Late_delivery_risk is {p_val}')
        print('---------')
    else:
        x=df_supply_chain[i][df_supply_chain['Late_delivery_risk']==1]
        y=df_supply_chain[i][df_supply_chain['Late_delivery_risk']==0]
        t_stat,p_val=stats.ttest_ind(x,y)
        print(f'p value for {i} and Late_delivery_risk is {p_val}')
        print('---------')
```

```
p value for Type and Late_delivery_risk is 5.128672571053333e-239
---------
p value for Category Name and Late_delivery_risk is 0.7179808169070767
---------
p value for Customer Segment and Late_delivery_risk is 0.5990740516327164
---------
p value for Market and Late_delivery_risk is 0.07044822520489222
---------
p value for Sales per customer and Late_delivery_risk is 0.1072215944768319
---------
p value for Delivery Status and Late_delivery_risk is 0.0
```

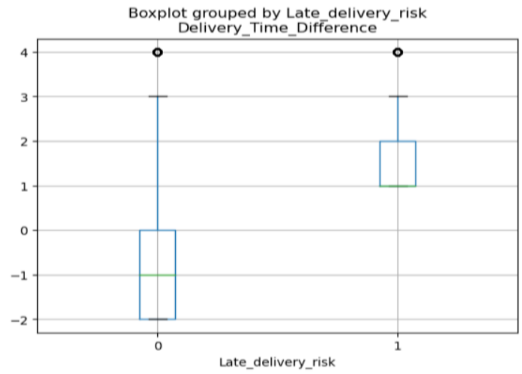
By calculating the p-value for each column with respect to the target variable we have selected the significant columns.

Feature selection for model building:

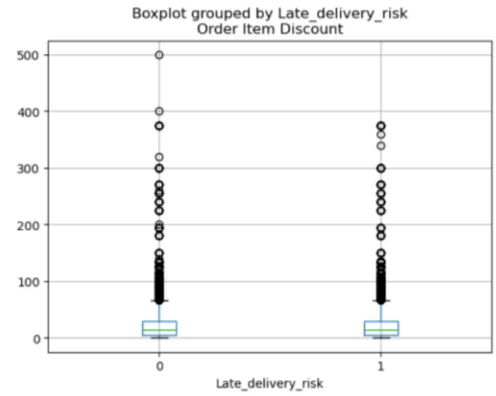
```
df_supply_chain.boxplot(column='Sales per customer',by='Late_delivery_risk')
plt.show()
```



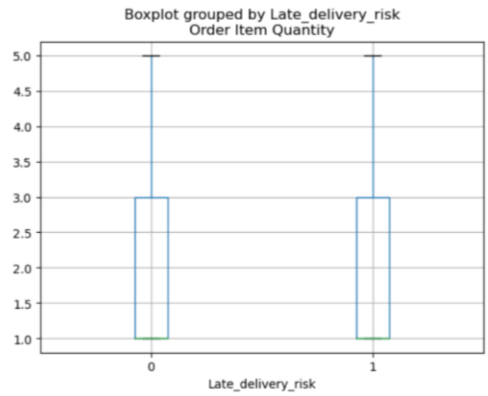

```
df_supply_chain.boxplot(column='Delivery_Time_Difference',by='Late_delivery_risk')
plt.show()
```



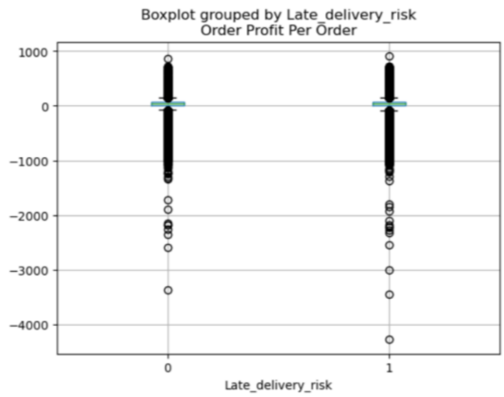

```
df_supply_chain.boxplot(column='Order Item Discount',by='Late_delivery_risk')
plt.show()
```



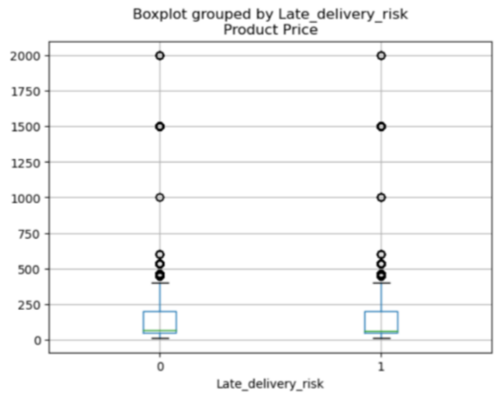

```
df_supply_chain.boxplot(column='Order Item Quantity',by='Late_delivery_risk')
plt.show()
```




```
df_supply_chain.boxplot(column='Order Profit Per Order',by='Late_delivery_risk')
plt.show()
```




```
df_supply_chain.boxplot(column='Product Price',by='Late_delivery_risk')
plt.show()
```



Observation:

From Box Plot we can infer that

1. Sales per customer,
2. Order item discount,
3. Order item quantity,
4. Order profit per order,
5. Product price

These columns exhibit little variation or substantial overlap. Thus the feature doesn’t have a strong impact on the target variable.

Encoding the categorical variables:

We are using Target Encoding. Target encoding also know as mean encoding, is a technique used in machine learning and predictive modeling to encode categorical variables by replacing them with the mean or other statistical measures of the target variable. The target variable represents the outcome or the variable being predicted.

```
from category_encoders import TargetEncoder
encoder = TargetEncoder()
df_supply_chain['Type'] = encoder.fit_transform(df_supply_chain['Type'], df_supply_chain['Delivery_Time_Difference'])
df_supply_chain['Customer Segment'] = encoder.fit_transform(df_supply_chain['Customer Segment'], df_supply_chain['Delivery_Time_D
df_supply_chain['Market'] = encoder.fit_transform(df_supply_chain['Market'], df_supply_chain['Delivery_Time_Difference'])
df_supply_chain['Order Country'] = encoder.fit_transform(df_supply_chain['Order Country'], df_supply_chain['Delivery_Time_Differe
df_supply_chain['Order Region'] = encoder.fit_transform(df_supply_chain['Order Region'], df_supply_chain['Delivery_Time_Differenc
df_supply_chain['Order State'] = encoder.fit_transform(df_supply_chain['Order State'], df_supply_chain['Delivery_Time_Difference'
df_supply_chain['Shipping Mode'] = encoder.fit_transform(df_supply_chain['Shipping Mode'], df_supply_chain['Delivery_Time_Differe
```

Model Building and Additional Data Treatments:

Classification Predictive Modelling

As originally mentioned in our objective, our aim with this analysis is to try and understand trends in order rejections. Accordingly, we will build a model that will help identify which of the features are most likely to affect rejections and consequently be useful in predicting rejections in the future.

As a start, we will build a logistic regression-based classifier. We shall use popular metrics such as precision and f1-score among other, to interpret the performance of our base model.

Separating our data into target and features sections and using the Train Test Split function to create appropriate data partitions to be feed into the model.

```
X = df_supply_chain[['Type','Delivery_Time_Difference','Customer Segment','Market','Order Country','Order Region','Order State','
y = df_supply_chain['Late_delivery_risk']
```

```
X.head(2)
```

	Type	Delivery_Time_Difference	Customer Segment	Market	Order Country	Order Region	Order State	Shipping Mode
0	0.557544	-1	0.564917	0.569365	0.580162	0.558235	0.570379	-0.004093
1	0.565062	1	0.564917	0.569365	0.615513	0.597465	0.698718	-0.004093

```
y.head(2)
```

```
0    0
1    1
Name: Late_delivery_risk, dtype: int64
```

```
# add a constant column to the dataframe
# while using the 'Logit' method in the Statsmodels library, the method do not consider the intercept by default
# we can add the intercept to the set of independent variables using 'add_constant()'
X = sm.add_constant(X)
```

```
# split data into train subset and test subset
# set 'random_state' to generate the same dataset each time you run the code
# 'test_size' returns the proportion of data to be included in the testing set
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state = 1, test_size = 0.2)
```

```
X_train.shape,X_test.shape,y_train.shape,y_test.shape
```

```
((144415, 9), (36104, 9), (144415,), (36104,))
```

Analysis and Scoring:

Now we have split our model in training and testing segments, and trained our model using the training segment, we shall use various methods to check performance.

First, we use the model.summary() function to output a table of various results from our Logistic Regression Classifier.

```
print(logreg.summary())
```

```
                           Logit Regression Results
==============================================================================
Dep. Variable:     Late_delivery_risk   No. Observations:               136625
Model:                          Logit   Df Residuals:                   136583
Method:                           MLE   Df Model:                           41
Date:                Thu, 18 May 2023   Pseudo R-squ.:                  0.2644
Time:                        13:30:41   Log-Likelihood:                -69074.
converged:                      False   LL-Null:                       -93895.
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================
```

```
LogReg=LogisticRegression()
LogReg.fit(X_train, y_train)
```

```
LogisticRegression()
```

```
LogReg.score(X_test, y_test)
```

```
0.7088737301285242
```

```
y_pred=LogReg.predict(X_test)
print (classification_report(y_test,y_pred))
```

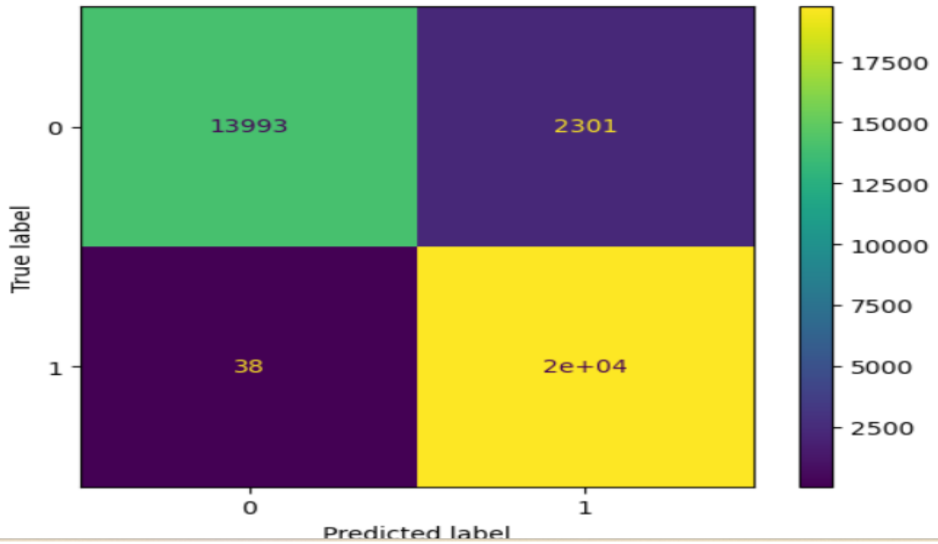
```
              precision    recall  f1-score   support

           0       0.63      0.88      0.73     15361
           1       0.85      0.57      0.68     18796

    accuracy                           0.71     34157
   macro avg       0.74      0.72      0.71     34157
weighted avg       0.75      0.71      0.70     34157
```

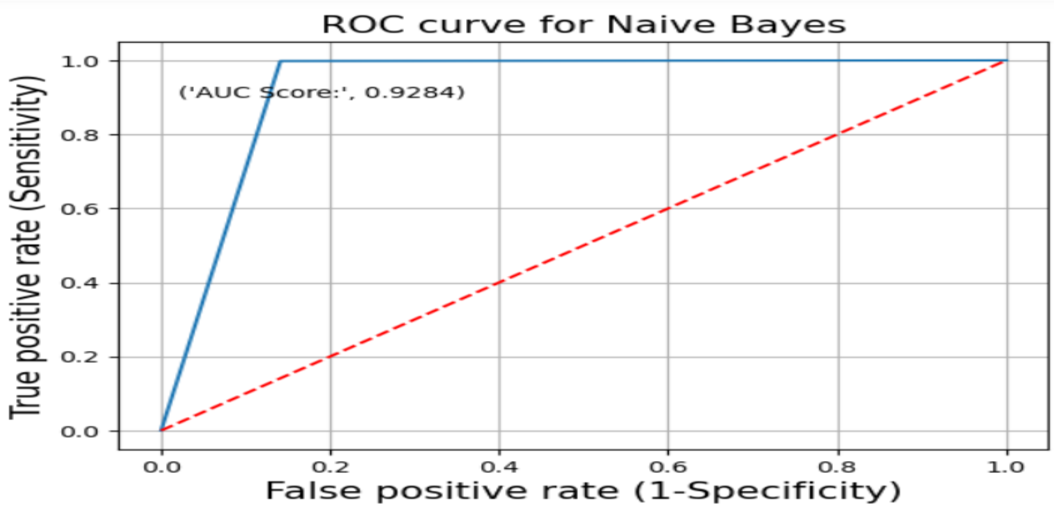
Logistic Regression has an accuracy percentage of 71%.

Naives Bayes Model:

```
nb_classifier = GaussianNB()
nb_classifier.fit(X_train,y_train)
y_pred_test = nb_classifier.predict(X_test)
y_pred_train = nb_classifier.predict(X_train)
nb_classifier.score(X_test,y_test)
print("Test Recall:", recall_score(y_test,y_pred_test))
print("Train Recall:", recall_score(y_train,y_pred_train))
ConfusionMatrixDisplay.from_predictions(y_test,y_pred_test)
plt.show()
```

```
Test Recall: 0.9980817768803635
Train Recall: 0.9981305341872245
```

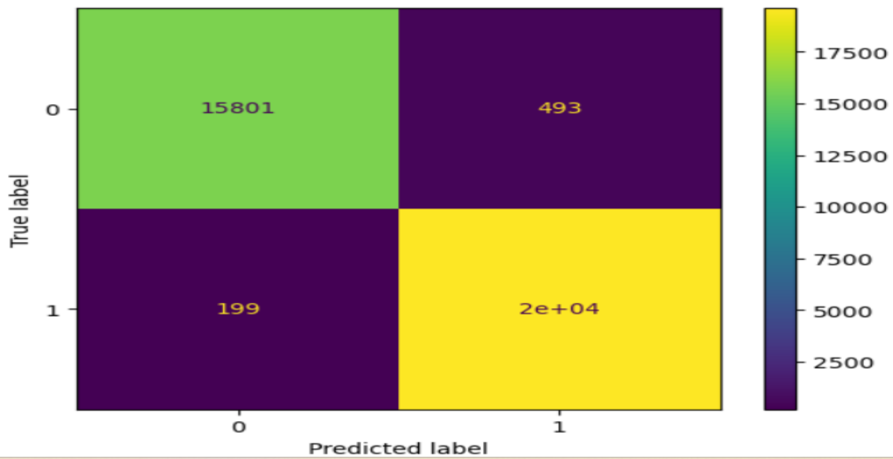




Decision Tree Classifier:

```
#Decision Tree classifier
clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)
y_pred_test = clf.predict(X_test)
y_pred_train = clf.predict(X_train)
print("Test recall", recall_score(y_test,y_pred_test))
print("Train recall", recall_score(y_train,y_pred_train))
ConfusionMatrixDisplay.from_predictions(y_test,y_pred_test)
plt.show()
```

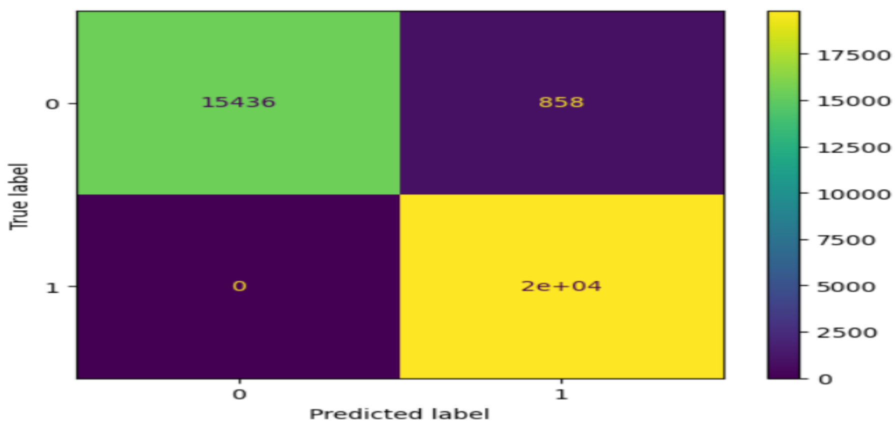
```
Test recall 0.9899545683997981
Train recall 0.9938105523766215
```

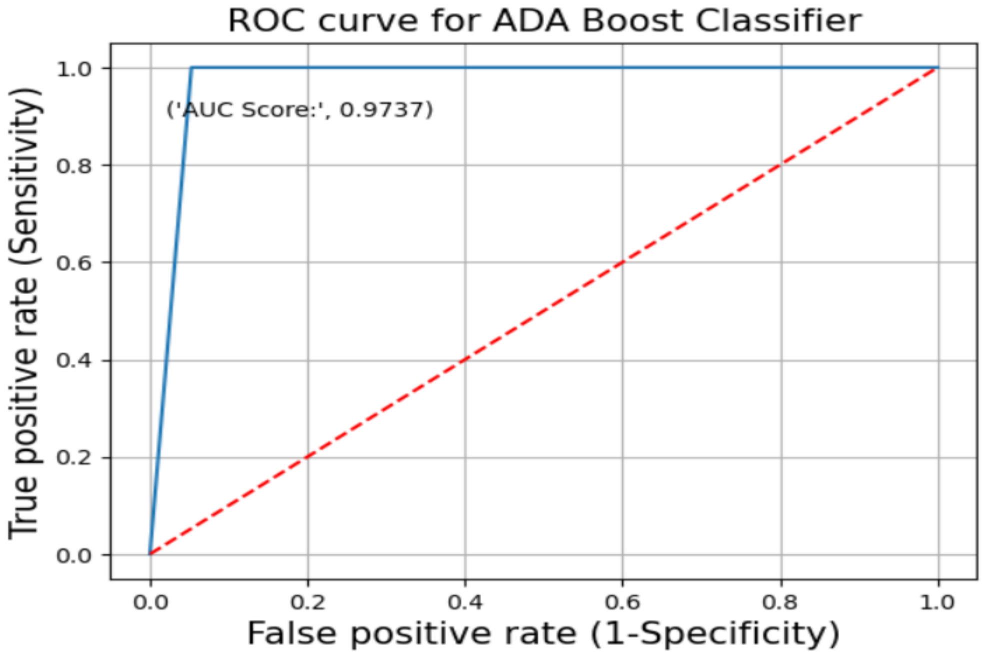


Ada Boost:

```
ad = AdaBoostClassifier()
ad.fit(X_train,y_train)
y_pred_test = ad.predict(X_test)
y_pred_train = ad.predict(X_train)
print("Test Recall", recall_score(y_test,y_pred_test))
print("Train Recall", recall_score(y_train,y_pred_train))
ConfusionMatrixDisplay.from_predictions(y_test,y_pred_test)
plt.show()
```

```
Test Recall 1.0
Train Recall 1.0
```

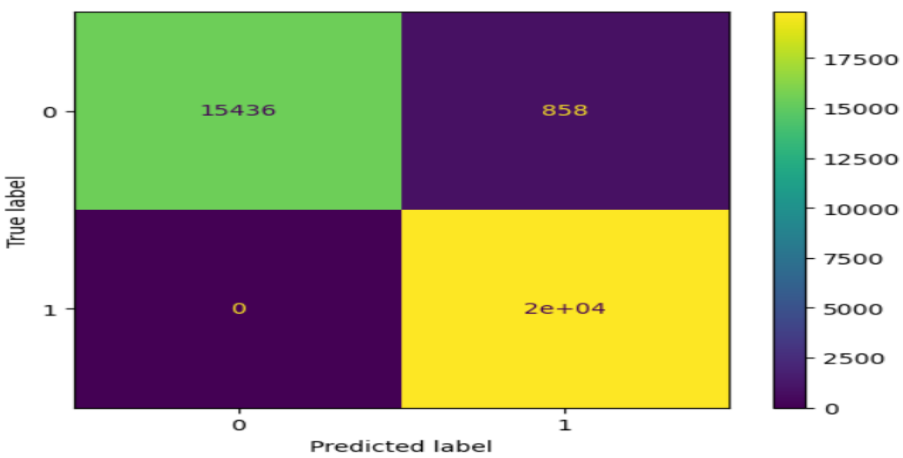


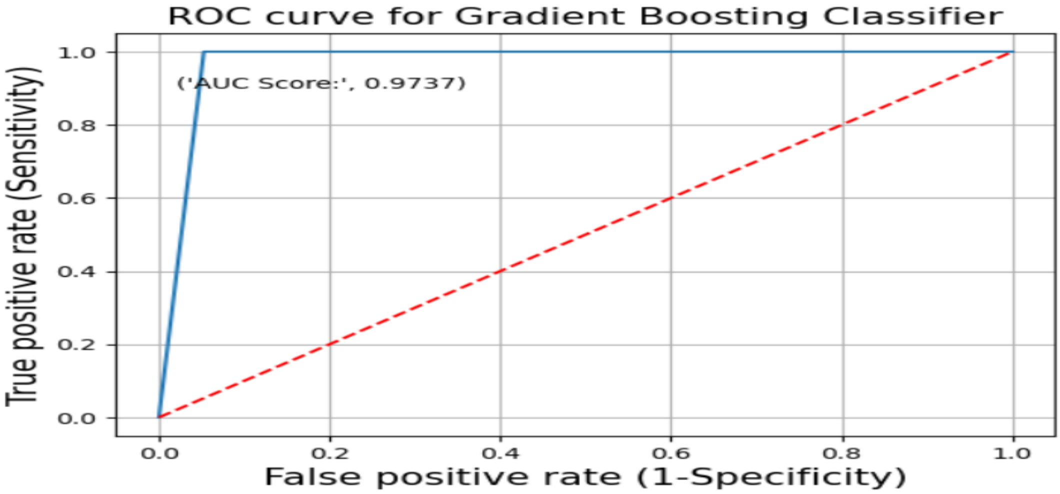


Gradient Boost Classifier:

```
GB = GradientBoostingClassifier()
GB.fit(X_train,y_train)
y_pred_test = GB.predict(X_test)
y_pred_train = GB.predict(X_train)
print("Test Recall", recall_score(y_test,y_pred_test))
print("Train Recall", recall_score(y_train,y_pred_train))
ConfusionMatrixDisplay.from_predictions(y_test,y_pred_test)
plt.show()
```

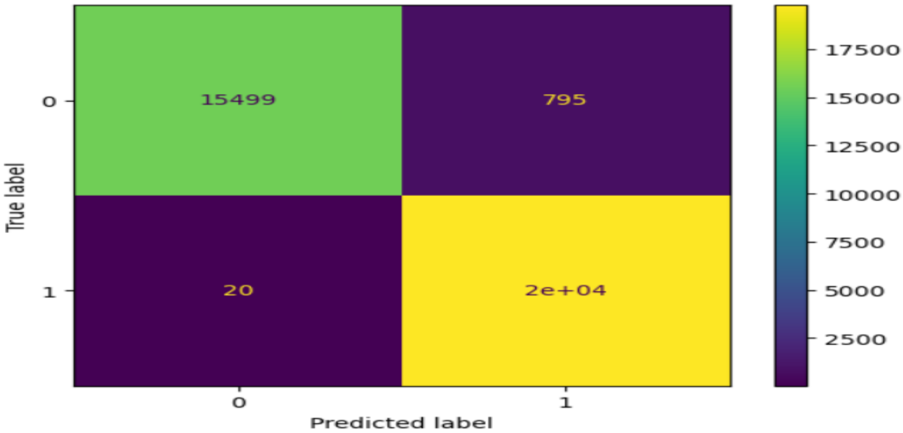
```
Test Recall 1.0
Train Recall 1.0
```

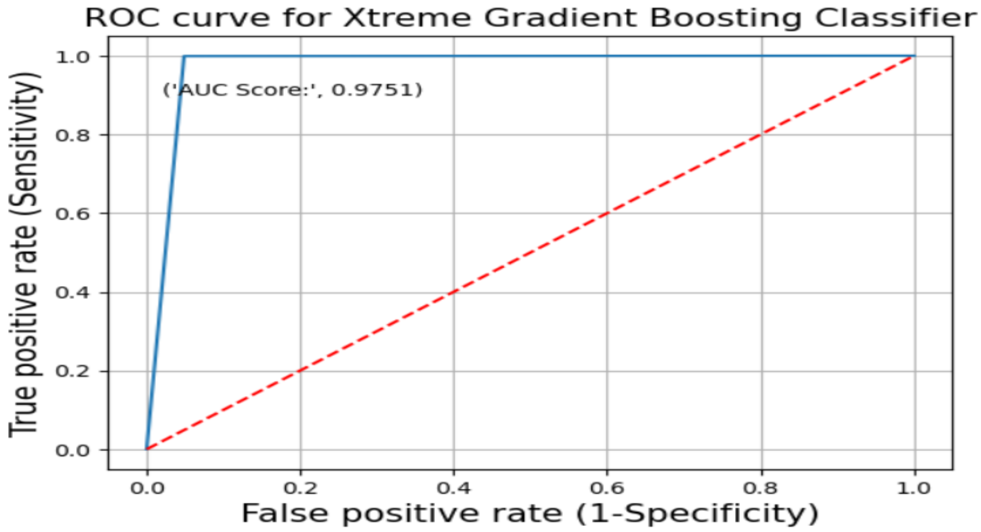




Xg Boost Classifier:

```
]: XGB = XGBClassifier()
   XGB.fit(X_train,y_train)
   y_pred_test = XGB.predict(X_test)
   y_pred_train = XGB.predict(X_train)
   print("Test Recall", recall_score(y_test,y_pred_test))
   print("Train Recall", recall_score(y_train,y_pred_train))
   ConfusionMatrixDisplay.from_predictions(y_test,y_pred_test)
   plt.show()

   Test Recall 0.9989904088844018
   Train Recall 0.9993684237119002
```

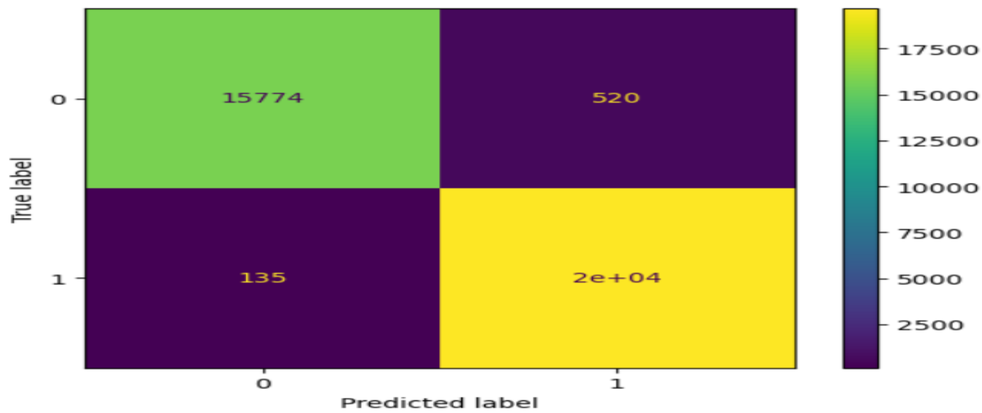




Random Forest:

```
rf = RandomForestClassifier()
rf.fit(X_train,y_train)
y_pred_test = rf.predict(X_test)
y_pred_train = rf.predict(X_train)
print("Test Recall", recall_score(y_test,y_pred_test))
print("Train Recall", recall_score(y_train,y_pred_train))
ConfusionMatrixDisplay.from_predictions(y_test,y_pred_test)
plt.show()
```

```
Test Recall 0.9931852599697123
Train Recall 0.9950989680043453
```



Comparative Analysis of all models:

	model name	accuracy	recall	precision	f1 score
4	Random Forest	0.981941	0.993236	0.974349	0.983702
3	Decision Tree	0.980889	0.990005	0.975527	0.982713
7	XGBoost	0.977426	0.998990	0.961380	0.979824
0	Logistic Regression	0.976235	1.000000	0.958487	0.978803
5	AdaBoost	0.976235	1.000000	0.958487	0.978803
6	Gradient Boosting	0.976235	1.000000	0.958487	0.978803
2	KNN	0.974740	0.984604	0.969819	0.977155
1	Naive Bayes	0.935215	0.998082	0.895755	0.944154

Based on the evaluation metrics, it can be concluded that the **Random Forest** model has performed well in predicting the target variable **'Late_delivery_risk'**. The model has achieved an overall accuracy of **98%** on the **test data**, which indicates that **99%** of the predictions made by the model were correct.

The precision of the model is high for both the classes, which means that the model has a low false positive rate. This is important as predicting that a delivery will be late when it is not, can result in unnecessary costs and damage to the reputation of the business.

The recall of the model is also good for both the classes, with a higher recall for the class 1, which indicates that the model has a low false negative rate. This is important as predicting that a delivery will not be late when it actually is, can result in dissatisfied customers and damage to the reputation of the business.

Business Interpretation:

The target variable in this model is **'Late_delivery_risk'**, which is an important metric for businesses involved in the delivery of goods to customers. Late deliveries can have a negative impact on customer satisfaction, which can ultimately harm the reputation of the business and result in lost revenue.

With the help of this model, businesses can identify the factors that are contributing to late deliveries and take corrective measures to improve the delivery process. The model can also be used to predict the risk of a delivery being late, which can help businesses prioritize their resources and take proactive steps to prevent delays.

For instance, based on the features included in the model, businesses can identify the customer segments that are most likely to experience late deliveries, the market segments that are most prone to delays, and the shipping modes that are most likely to result in late deliveries. This information can be used to optimize the delivery process and reduce the risk of late deliveries.

In addition, the model can also be used to identify the regions or states where late deliveries are most common. This can help businesses to focus their efforts on these areas and take steps to improve the delivery process such as increasing the number of delivery personnel or improving the transportation infrastructure.

Overall, the model can provide valuable insights to businesses involved in the delivery of goods and help them improve the efficiency of their delivery process, which can ultimately lead to increased customer satisfaction and improved business performance.

Limitations of the model:

Limited feature set: The model uses a limited set of features to predict the target variable 'Late_delivery_risk'. There may be other important factors that contribute to the risk of late delivery, which are not included in the model. Therefore, the model may not capture the full complexity of the problem.

Lack of temporal data: The model does not take into account the temporal nature of the data. Delivery patterns and customer behavior may change over time, and the model may not be able to capture these changes.

Lack of external factors: The model does not take into account external factors that may impact the delivery process, such as weather conditions, traffic congestion, or transportation disruptions. Therefore, the model may not be able to fully capture the complexity of the problem and may lead to inaccurate predictions.

Model Interpretability: Lack of transparency i.e. Random Forest models are often referred to as "black box" models because they are difficult to interpret and understand compared to simpler models like linear regression. The ensemble nature of Random

Forest, where multiple decision trees are combined, makes it challenging to extract meaningful insights from the model's internal workings.

Limited visualization i.e. Random Forest models are not easily visualized due to their complexity. Unlike decision trees, which can be visualized as a tree-like structure, Random Forests do not lend themselves to intuitive visual representations. This makes it difficult to explain the decision-making process of the model to non-technical stakeholders.