greatlearning
Learning for Life

GROUP 2

Capstone Project – Final Report

Late Delivery Status Prediction

Mentor: Mr. Animesh Tiwari

Thanai Pouldas

Sairam Boga

Harshitha Dodla

Dheeraj Doddi

Gnanendra kumar

greatlearning
Learning for Life

CONTENT

INTRODUCTION

PROBLEM STATEMENT:

We have been given a total of 53 attributes, these attributes contain worldwide product supply data.
Analyzing the world wide supply chain dataset, to predict if a product will be delivered Late or not and also predicting the probable Sales for the product.

PROJECT OUTCOME:

Delivering a product on time is an important and necessary commitment given to the customer in this field of Supply Chain Industry. If the company fails to deliver the product by scheduled time, there is a high chance of losing the customer for future transactions.
This project helps the business client to predict whether the order would be shipped late or not and also predicts the sales.

INDUSTRY REVIEW:

- **Purpose** – The field of supply chain management (SCM) has historically been informed by knowledge from narrow functional areas. While some effort towards producing a broader organizational perspective has been made, nonetheless, SCM continues to be largely eclectic with little consensus on its conceptualization and research methodological bases. This paper seeks to clarify aspects of this emerging perspective.
- Supply Chain Management is a network of facilities that produce raw materials, transform them into intermediate goods and then final products, and deliver the products to customers through a distribution system. It spans procurement, manufacturing and distribution (Lee & Billington 1995) the basic objective of supply chain management is to “optimize performance of the chain to add as much value as possible for the least cost possible”. In other words, it aims to link all the supply chain
- Agents to jointly cooperate within the firm as a way to maximize productivity in the supply chain and deliver the most benefits to all related parties (Finch 2006). Adoption of Supply chain management practices in industries has steadily increased since the 1980s. A number of definitions are proposed and the concept is discussed from many perspectives. However, Cousins et al. (2006); Sachan and

Datta (2005); Storey et al. (2006) provided an excellent review on supply chain management literature. These papers define the concept, principals, nature, and development of SCM and indicate that there is an intense research being conducted around the world in this field that critically assessed developments in the theory and practice of supply management.

DATA SET INFORMATION

Original Owners of Database:

"DataCo SMART SUPPLY CHAIN FOR BIG DATA ANALYSIS",
Mendeley Data, V5.
Donors of database:
1.Constante, Fabian;
2.Silva, Fernando;
3.Pereira, António (2019).

Relevant Papers:

- Kumar Shukla et al. / International Journal of Engineering Science and Technology (IJEST)
- International Journal of Operations & Production Management Vol. 26 No. 7, 2006 pp. 703-729 q Emerald Group Publishing Limited 0144-3577 DOI 10.1108/01443570610672202
- Cousins et al. (Rajendra 2006); Sachan and Datta (2005); Storey et al. (2006)

DATA SET AND DOMAIN:

- A dataset is a collection of data, and it can be structured or unstructured.
- A structured data is represented in a tabular format, where every column of the table represents a particular variable, and each row corresponds to a given record of the dataset in question.
- A DataSet of Supply Chains by the company DataCo Global was used for analysis in this report. This Dataset of Supply Chain allows the use of Machine Learning Algorithms.
- **Areas of important registered activities** : Provisioning , Production , Sales , Commercial Distribution.It also allows the correlation of Structured Data with Unstructured Data for knowledge generation.

DATA DESCRIPTION:

The data set consists of 180519 observations and 53 features before the cleaning and contains information for delivered status, category names, includes information such as when order was placed, customer id, the total sales per customer, order status, discount amount, type of product, etc.

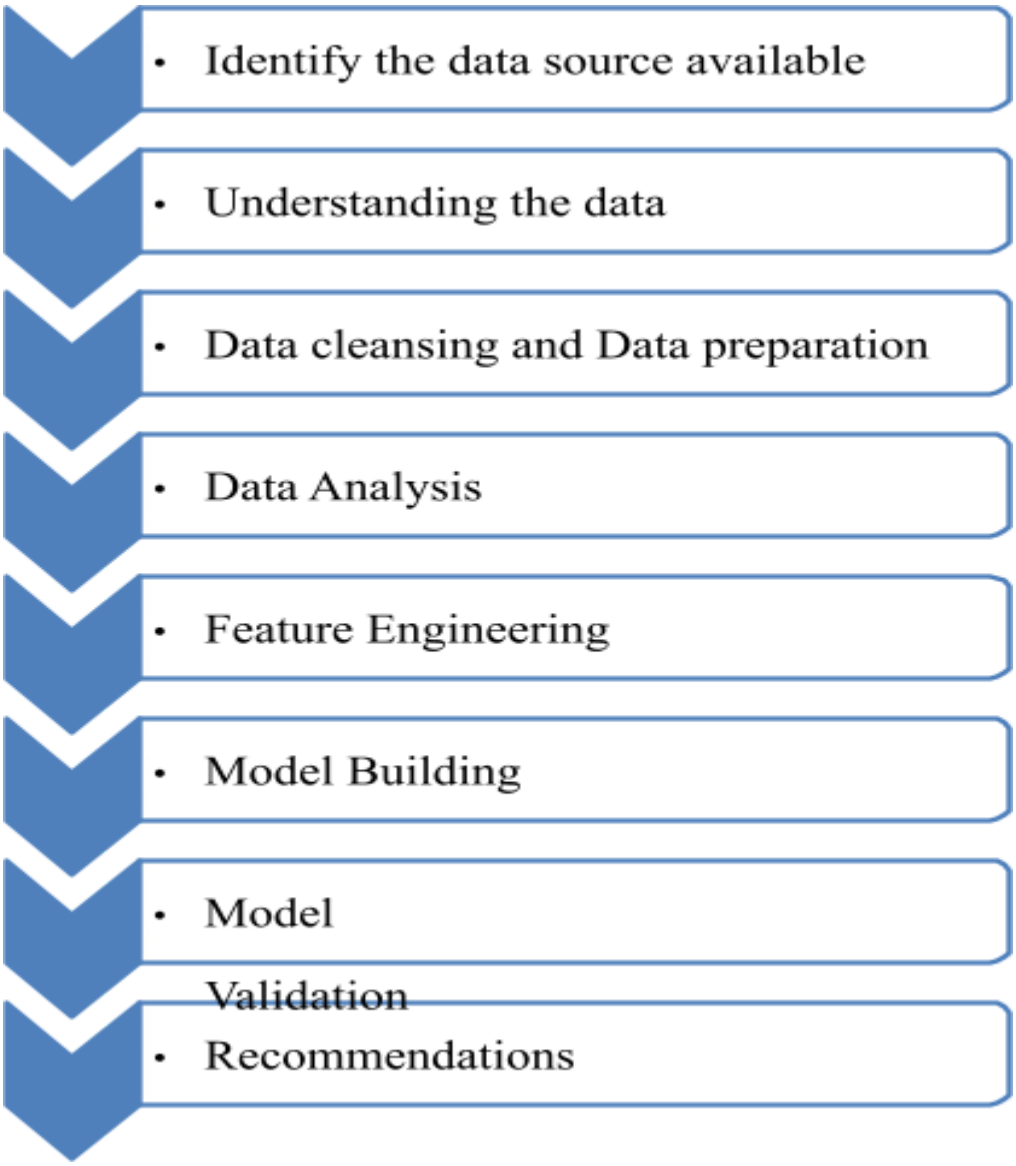
SL No	Variables	Variable Information	Data Type
1	**Type**	Type of transaction made	Categorical
2	**Days for shipping (real)**	Actual shipping days of the purchased product	Numerical
3	**Days for shipment (scheduled)**	Days of scheduled delivery of the purchased product	Numerical
4	**Benefit per order**	Earnings per order placed	Numerical
5	**Sales per customer**	Total sales made per customer	Numerical
6	**Delivery Status**	Delivery status of orders: Advance shipping , Late delivery , Shipping canceled , Shipping on time	Object
7	**Late_delivery_risk**	Categorical variable that indicates if sending is late (1), it is not late (0)	Numerical
8	**Category Id**	Product category code	Numerical

9	**Category Name**	Description of the product category	Object
10	**Customer City(Source City)**	City where the customer made the purchase	Object
11	**Customer Country (Source Country)**	Country where the customer made the purchase	Object
12	**Customer Email**	Customer's email	Object
13	**Customer Fname**	Customer name	Object
14	**Customer Id**	Customer ID	Numerical
15	**Customer Lname**	Customer lastname	Object
16	**Customer Password**	Masked customer key	Object
17	**Customer Segment**	Types of Customers: Consumer , Corporate , Home Office"	Object
18	**Customer State (Source State)**	State to which the store where the purchase is registered belongs	Object
19	**Customer Street**	Street to which the store where the purchase is registered belongs	Object
20	**Customer Zipcode**	Customer Zipcode	Numerical
21	**Department Id**	Department code of store	Numerical
22	**Department Name**	Department name of store	Object
23	**Latitude**	Latitude corresponding to location of store	Numerical
24	**Longitude**	Longitude corresponding to location of store	Numerical
25	**Market**	Market to where the order is delivered : Africa , Europe , LATAM , Pacific Asia , USCA"	Object
26	**Order City (Destination city)**	Destination city of the order	Object
27	**Order Country (Destination Country)**	Destination country of the order	Object
28	**Order Customer Id**	Customer order code	Numerical
29	**order date (DateOrders)**	Date on which the order is made	Object
30	**Order Id**	Order code	Numerical
31	**Order Item Cardprod Id**	Product code generated through the RFID reader	Numerical

32	**Order Item Discount**	Order item discount value	Numerical
33	**Order Item Discount Rate**	Order item discount percentage	Numerical
34	**Order Item Id**	Order item code	Numerical
35	**Order Item Product Price**	Price of products without discount	Numerical
36	**Order Item Profit Ratio**	Order Item Profit Ratio	Numerical
37	**Order Item Quantity**	Number of products per order	Numerical
38	**Sales**	Value in sales	Numerical
39	**Order Item Total**	Total amount per order	Numerical
40	**Order Profit Per Order**	Order Profit Per Order	Numerical
41	**Order Region (Destination 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	Object
42	**Order State(Destination State)**	State of the region where the order is delivered	Object
43	**Order Status**	Order Status : COMPLETE , PENDING , CLOSED , PENDING_PAYMENT ,CANCELED , PROCESSING ,SUSPECTED_FRAUD ,ON_HOLD ,PAYMENT_REVIEW"	Object
44	**Order Zipcode**	Destination zipcode	Numerical
45	**Product Card Id**	Product code	Numerical
46	**Product Category Id**	Product category code	Numerical
47	**Product Description**	Product Description	Numerical
48	**Product Image**	Link of visit and purchase of the product	Object
49	**Product Name**	Product Name	Object
50	**Product Price**	Product Price	Numerical
51	**Product Status**	Status of the product stock :If it is 1 not available , 0 the product is available	Numerical

52	**Shipping date**	Exact date and time of shipment	Object
53	**Shipping Mode,**	The following shipping modes are Standard Class,First Class,Second Class, Same Day"	Object

PROJECT METHODOLOGY



PRE-PROCESSING DATA ANALYSIS

Data Preparation:

Data preprocessing is a crucial step that helps enhance the quality of data to promote the extraction of meaningful insights from the data. Data preprocessing in Machine Learning refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for building and training Machine Learning models.

Data Preparation is the process of collecting, cleaning, and consolidating data into one file or data table, primarily for use in analysis.

Missing/Null Values:

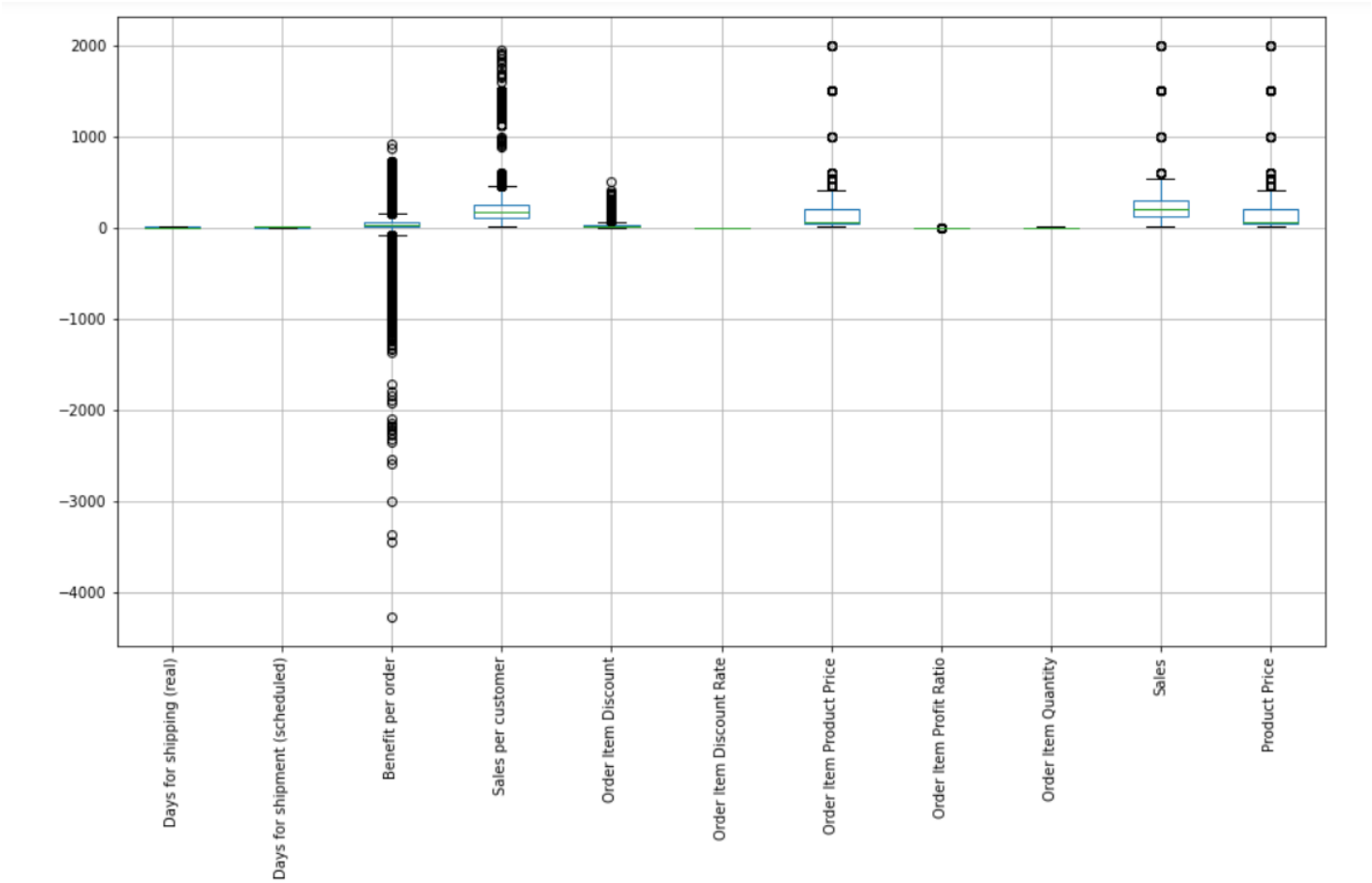
Impute or drop features with missing values based on the percentage of missing values and relevance for model building.

Order Zip Code has 86% and Product Description has 100% of Null values and these features are not useful for our further analysis so we are dropping these columns.

Type	0
Days for shipping (real)	0
Days for shipment (scheduled)	0
Benefit per order	0
Sales per customer	0
Delivery Status	0
Late_delivery_risk	0
Category Id	0
Category Name	0
Customer City	0
Customer Country	0
Customer Email	0
Customer Fname	0
Customer Id	0
Customer Lname	8
Customer Password	0
Customer Segment	0
Customer State	0
Customer Street	0
Customer Zipcode	3
Department Id	0
Department Name	0
Latitude	0
Longitude	0
Market	0
Order City	0
Order Country	0
Order Customer Id	0
order date (DateOrders)	0
Order Id	0

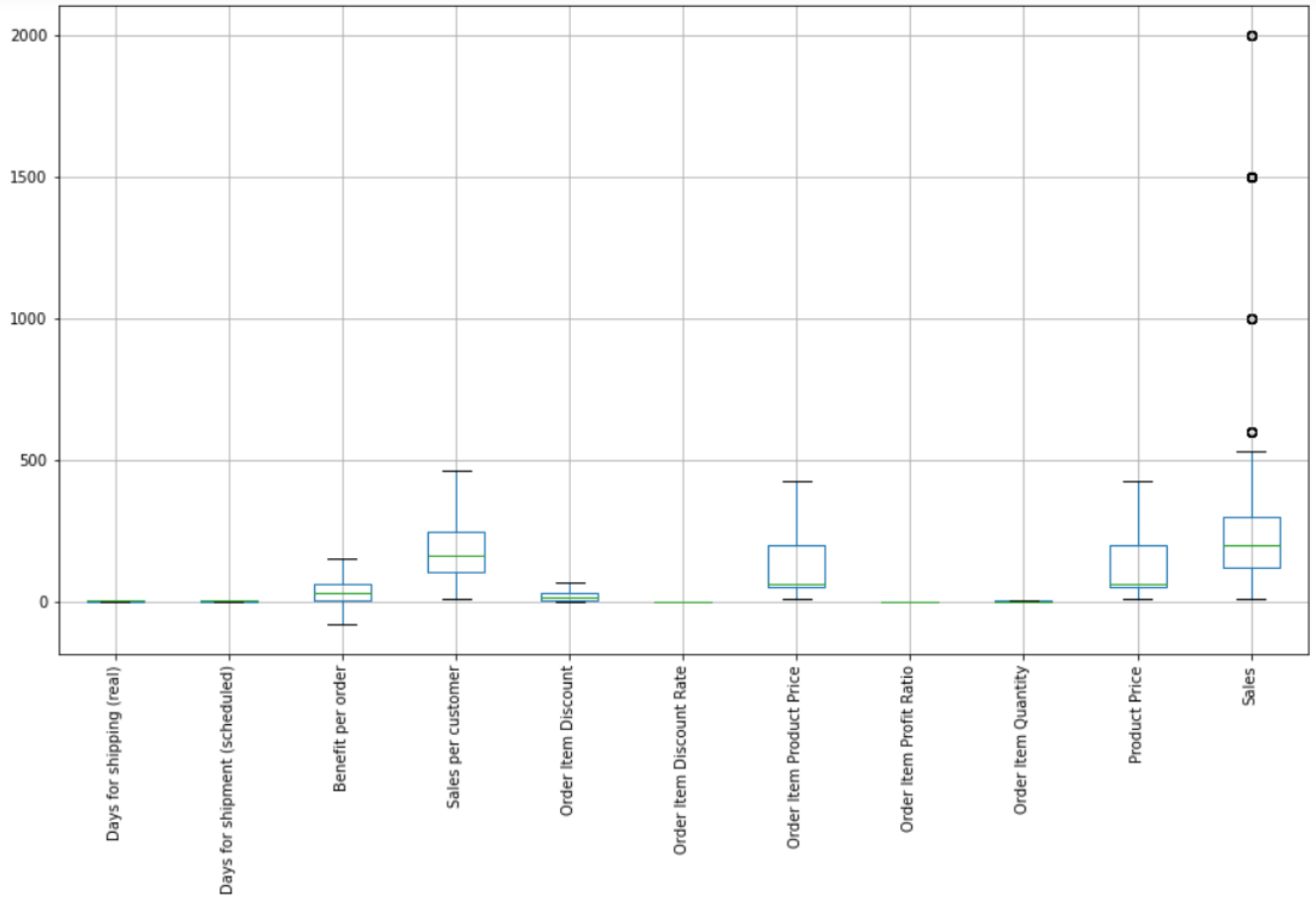
Order Item Product Price	0
Order Item Profit Ratio	0
Order Item Quantity	0
Sales	0
Order Item Total	0
Order Profit Per Order	0
Order Region	0
Order State	0
Order Status	0
Order Zipcode	155679
Product Card Id	0
Product Category Id	0
Product Description	180519
Product Image	0
Product Name	0
Product Price	0
Product Status	0
shipping date (DateOrders)	0
Shipping Mode	0

OUTLIERS:



The outliers are present in the Benefit per order,Sales per customer,Order Item Discount,Order Item product price,order item profit ratio,Product price. As Sales is one of our target variables we are not excluding the outliers in this.

Here we treat null values using the winsorization technique(Capping).



After treating the extreme outliers here's the boxplot of those features.

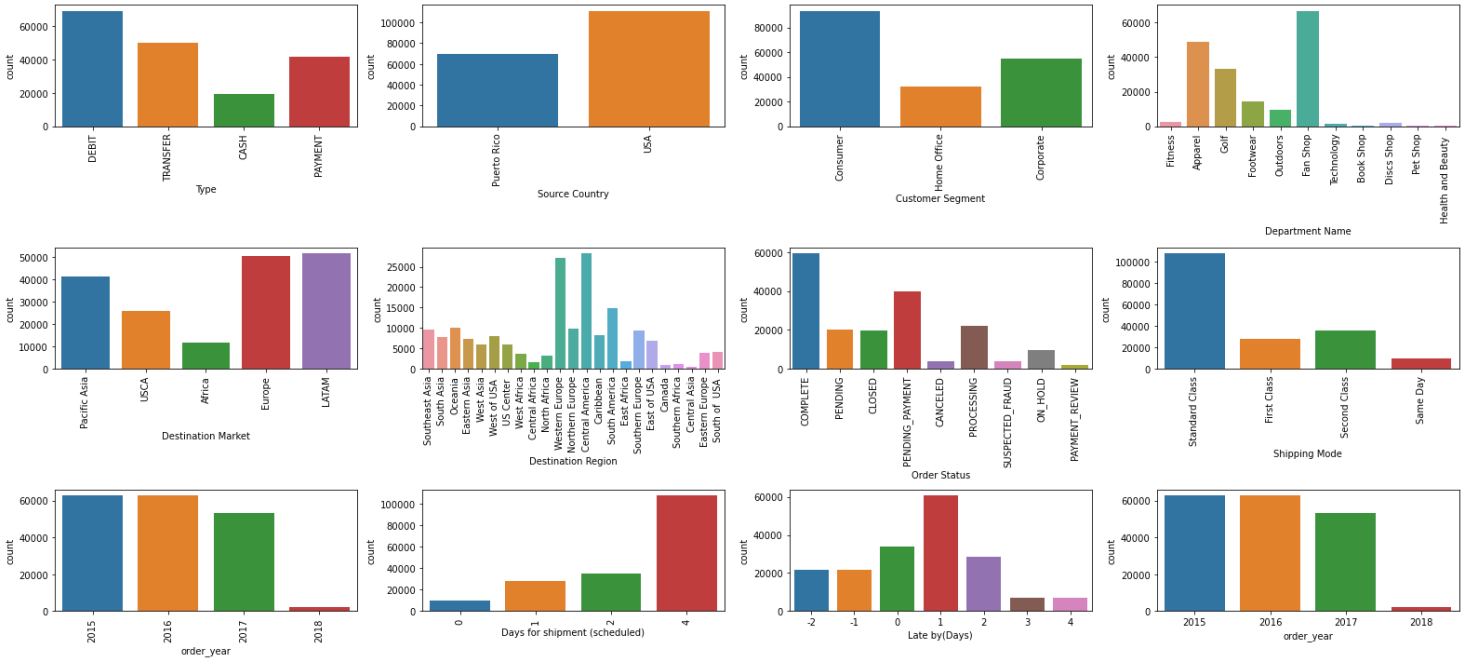
REDUNDANT COLUMNS:

The below are the redundant features that are dropped from the dataset.

- 'Category Id','Customer Email','Customer Fname','Customer Id','Customer Lname','Customer Password','Customer Zipcode','Customer Street','Department Id','Latitude','Longitude','Order Customer Id','Order Id','Order Item Cardprod Id',
- 'Order Item Total','Order Item Id','Order Profit Per Order','Order Zipcode','Product Image','Product Card Id','Product Category Id','Product Description','Product Status'

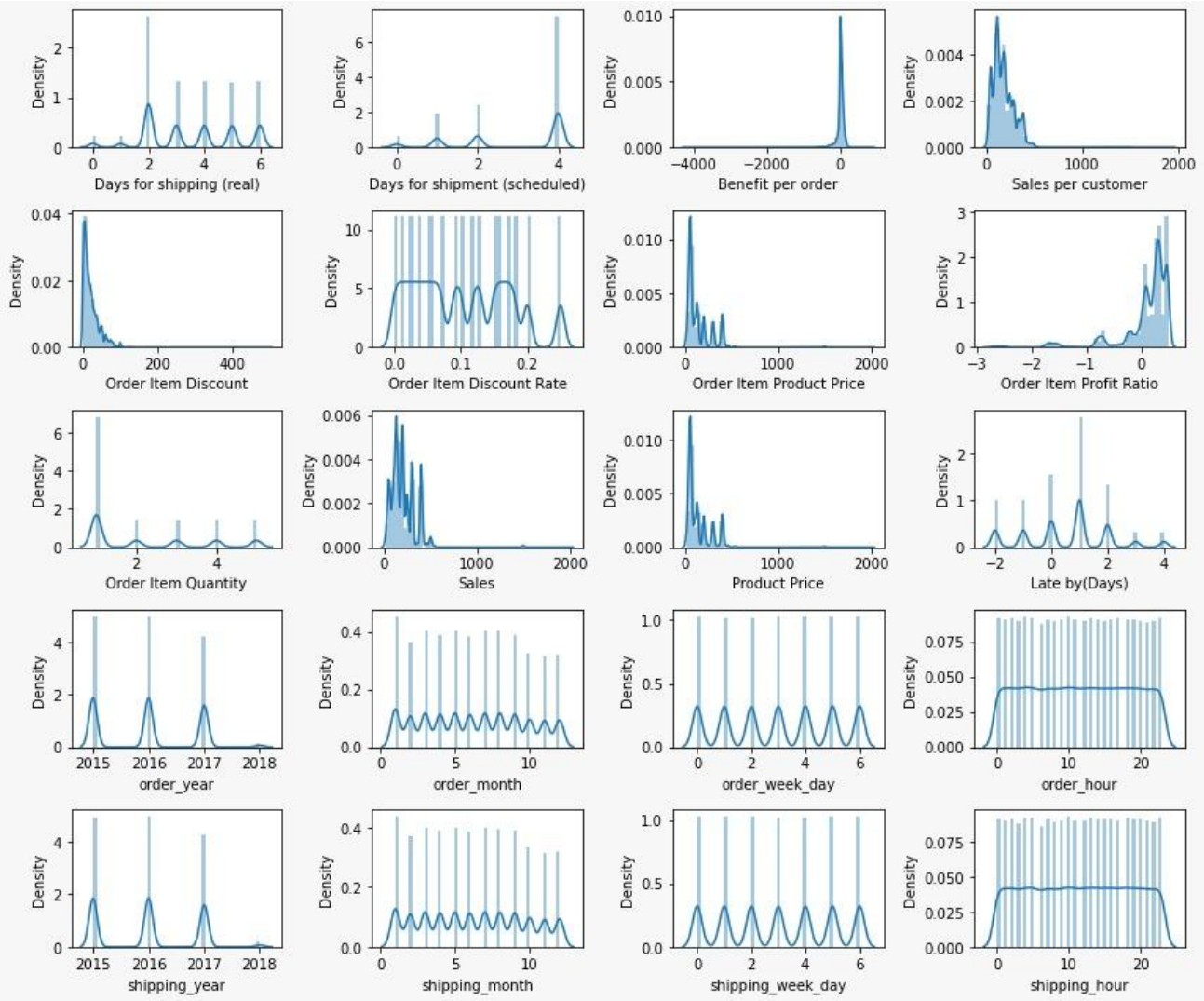
EXPLORATORY DATA ANALYSIS & BUSINESS INSIGHTS

UNIVARIATE ANALYSIS FOR CATEGORICAL VARIABLES:



- For the variable Type we can observe that most of the customers chose debit as payment type with cash being least prefered.
- For the variable Delivery status the count of late deliveries is high.
- Late_delivery_risk is our target variable and the data in the variable is not imbalanced.
- The USA is the most sourced country followed by Puerto Rico.
- Most of the customers are consumers.
- Department Fan Shop is the most procured.
- Most products are shipped to Europe and Latin America.
- Most of the customers preferred ‘Standard Class’ as their shipping mode.
- Among the late delivered products, most of them were late by ONE day
- We can see that there was a slight decrease in the orders placed in year 2017 compared to previous years

UNIVARIATE ANALYSIS FOR NUMERICAL VARIABLES:

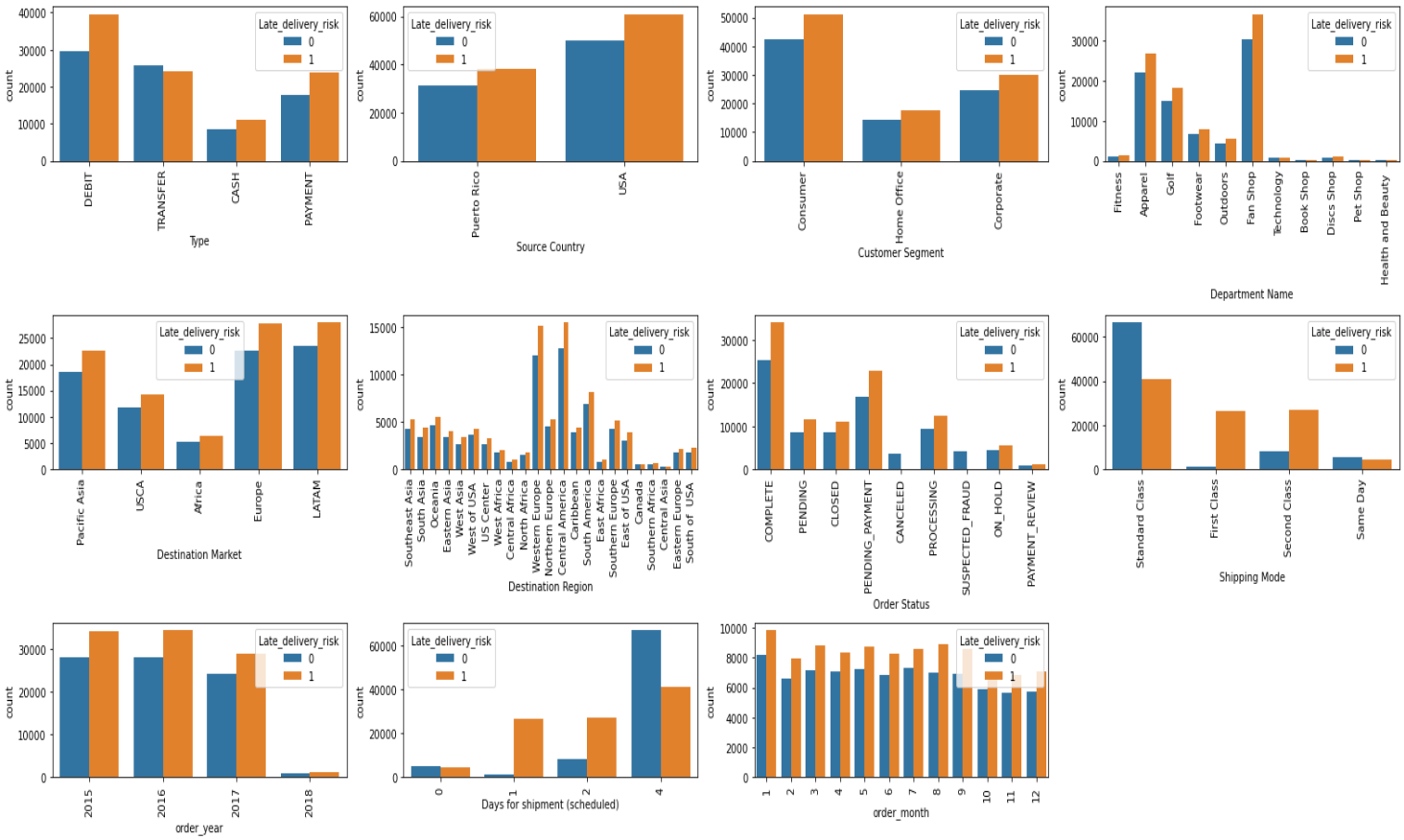


- We can see that the data is not normally distributed in any of the numerical variables. We can make the data normal slightly by transforming using any of the transformations i.e log ,power transformation.

BIVARIATE ANALYSIS

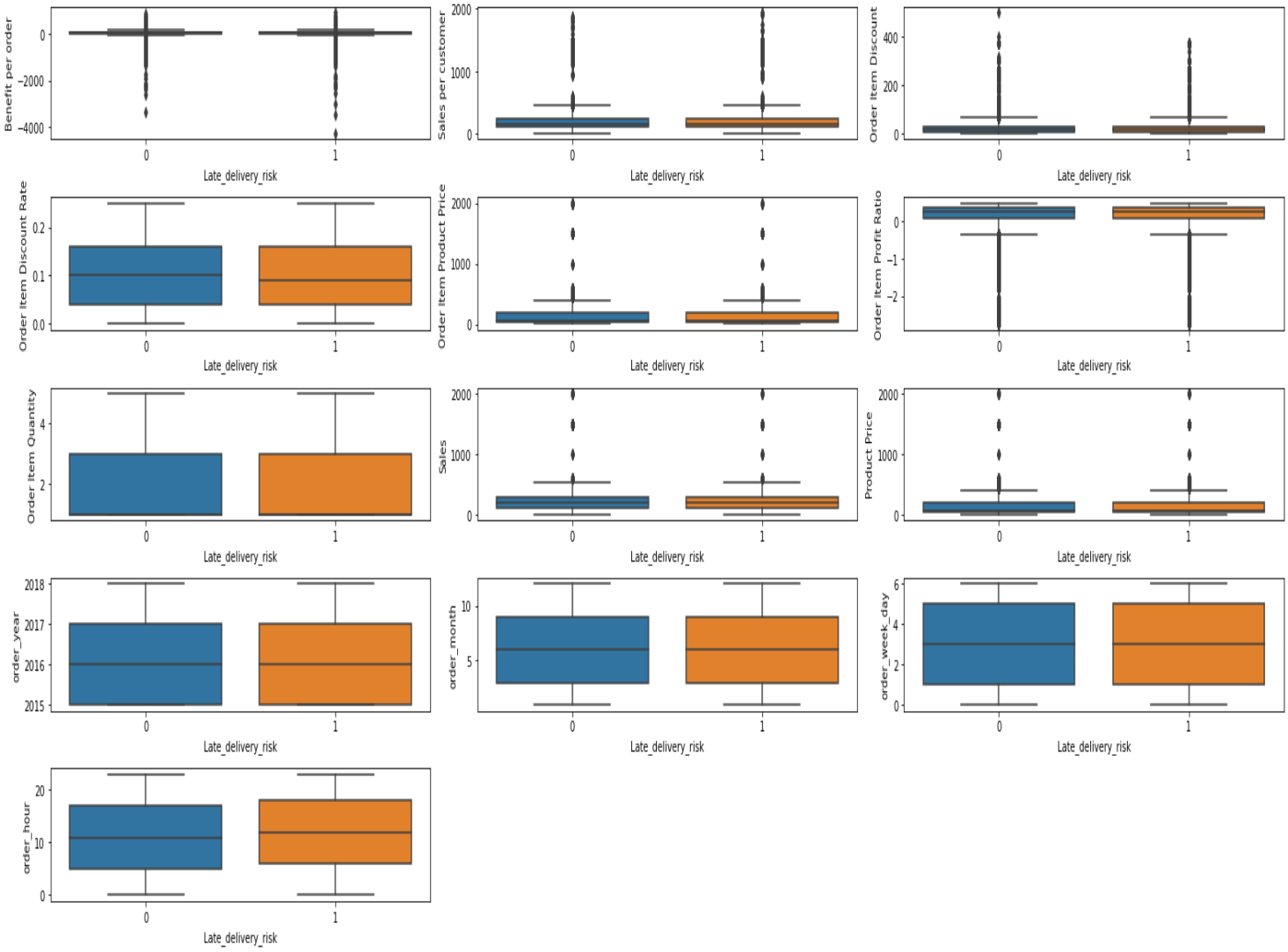
RELATIONSHIP WITH TARGET VARIABLE(Late_delivery_risk):

Categorical independent variables vs target variable(Late_delivery_risk):



- We can observe that late deliveries are prominent in all the types of payments except Transfer Payments.
- When the days for scheduled shipment was 4 days, then most of the products were delivered either on time or before scheduled time.
- In all other variables for the various conditions we can observe that most of the products are delivered late.

Numerical independent variables vs target variable(Late_delivery_risk)



- We can observe that the means of the data is similar for both products delivered late and the ones which are not delivered late, for all of the numerical variables.
- We can prove this statistically by using tests which are mentioned at a later stage in this report.

STATISTICAL TESTS TO VALIDATE THE OBSERVATION MADE FROM EDA FOR TARGET VARIABLE LATE DELIVERY RISK

We have made a few observations from graphs. We can further validate these observations using statistical tests like Chi-Square Test for Independence,Anova,Kruskal Wallis Test.

Chi-Square Test for Independence

This test is used to test whether the categorical feature are independent or not with respect to the Target Categorical Variable(Late_delivery_risk).

*H*0 : The variables are independent.

*H*1 : The variables are not independent (i.e. variables are dependent).

```
For feature  Type   p-value: 0.0
For feature  Delivery Status   p-value: 0.0
For feature  Category Name   p-value: 0.716
For feature  Source City   p-value: 0.0
For feature  Source Country   p-value: 0.65
For feature  Customer Segment   p-value: 0.593
For feature  Source State   p-value: 0.0
For feature  Department Name   p-value: 0.758
For feature  Destination Market   p-value: 0.07
For feature  Destination City   p-value: 0.0
For feature  Destination Country   p-value: 0.0
For feature  order date (DateOrders)   p-value: 0.0
For feature  Destination Region   p-value: 0.0
For feature  Destination State   p-value: 0.0
For feature  Order Status   p-value: 0.0
For feature  Product Name   p-value: 0.815
For feature  shipping date (DateOrders)   p-value: 0.0
For feature  Shipping Mode   p-value: 0.0
For feature  customer full name   p-value: 0.0
```

- we can observe that for category name, source country, customer segment,Department name,product name the p-value is greater than 0.05. Thus we fail to reject (i.e. accept) the null hypothesis and conclude that these variables are independent with respect to target variable late delivery risk
- For the remaining variables p-value is less than 0.05, so we reject the null hypothesis and conclude that these variables are dependent with respect to the target variable.

Shapiro-Wilk Test:

The Shapiro-Wilk Test is used to check the normality of the data.

H0(null hypothesis):The data was drawn from a normal distribution.
H1: The data was not drawn from a normal distribution

```
Shapiro Test p value for Days for shipping (real) is 0.0
Shapiro Test p value for Days for shipment (scheduled) is 0.0
Shapiro Test p value for Benefit per order is 0.0
Shapiro Test p value for Sales per customer is 0.0
Shapiro Test p value for Order Item Discount is 0.0
Shapiro Test p value for Order Item Discount Rate is 0.0
Shapiro Test p value for Order Item Product Price is 0.0
Shapiro Test p value for Order Item Profit Ratio is 0.0
Shapiro Test p value for Order Item Quantity is 0.0
Shapiro Test p value for Sales is 0.0
Shapiro Test p value for Product Price is 0.0
Shapiro Test p value for Late by(Days) is 0.0
Shapiro Test p value for order_year is 0.0
Shapiro Test p value for order_month is 0.0
Shapiro Test p value for order_week_day is 0.0
Shapiro Test p value for order_hour is 0.0
Shapiro Test p value for shipping_year is 0.0
Shapiro Test p value for shipping_month is 0.0
Shapiro Test p value for shipping_week_day is 0.0
Shapiro Test p value for shipping_hour is 0.0
```

- As we can see that the shapiro test has failed for the data and the Data is not normally distributed, we proceed to doing a Non-Parametric test, instead of One way ANOVA.

Kruskal Wallis test:

It is used to check the equality of population medians for more than two independent samples. It is a non-parametric test that works similar to ANOVA, but doesn't require the tests of normality and equal variances to be satisfied.

The null and alternative hypothesis is given as:

Ho:The median of all treatments are the same.

H1: At least one treatment has a different median.

To understand the impact of the numerical variables on Late Delivery Risk we conducted a Kruskal Wallis test. If the result of the test indicates dependency (reject the null hypothesis) we can state that there is some impact on the target variable.

```
for i in df_num.columns:
    t0=df[df['Late_delivery_risk']==0][i]
    t1=df[df['Late_delivery_risk']==1][i]
    st,pval=stats.kruskal(t0,t1)
    print('Kruskal Test p value for',i,'is',round(pval,3))
```

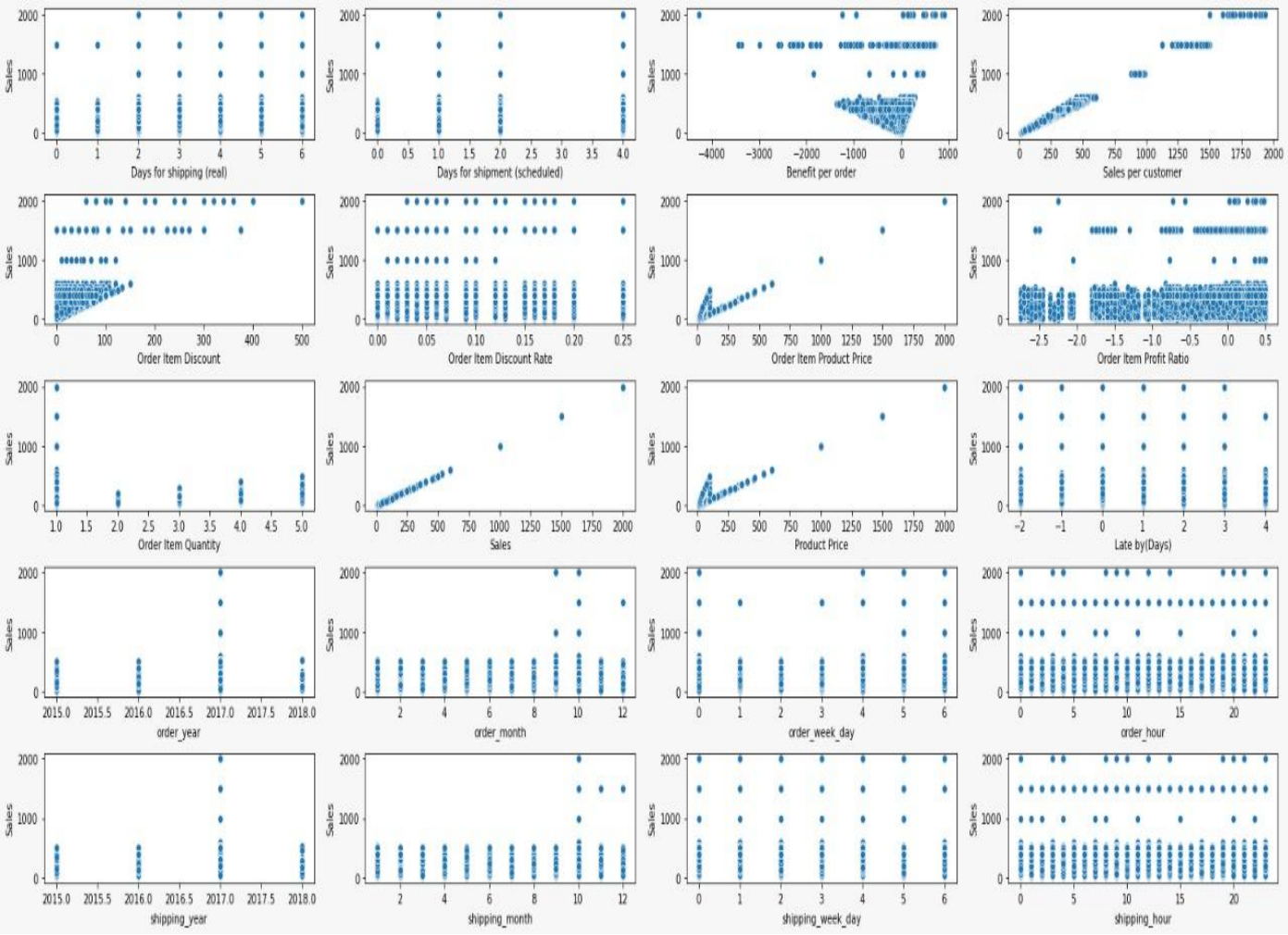
```
Kruskal Test p value for Benefit per order is 0.209
Kruskal Test p value for Sales per customer is 0.261
Kruskal Test p value for Order Item Discount is 0.37
Kruskal Test p value for Order Item Discount Rate is 0.876
Kruskal Test p value for Order Item Product Price is 0.282
Kruskal Test p value for Order Item Profit Ratio is 0.673
Kruskal Test p value for Order Item Quantity is 0.812
Kruskal Test p value for Sales is 0.231
Kruskal Test p value for Product Price is 0.282
Kruskal Test p value for order_year is 0.317
Kruskal Test p value for order_month is 0.272
Kruskal Test p value for order_week_day is 0.633
Kruskal Test p value for order_hour is 0.0
Kruskal Test p value for Days for shipment (scheduled) is 0.0
```

- we can observe that for Days for shipment(Scheduled),Order hour the p-value is lesser than 0.05. Thus we reject the null hypothesis and conclude that The median of all treatments are not the same.
- For the remaining variables p-value is greater than 0.05, so we fail to reject the null hypothesis(i.e. accept) and conclude that The median of all treatments are the same.

BIVARIATE ANALYSIS

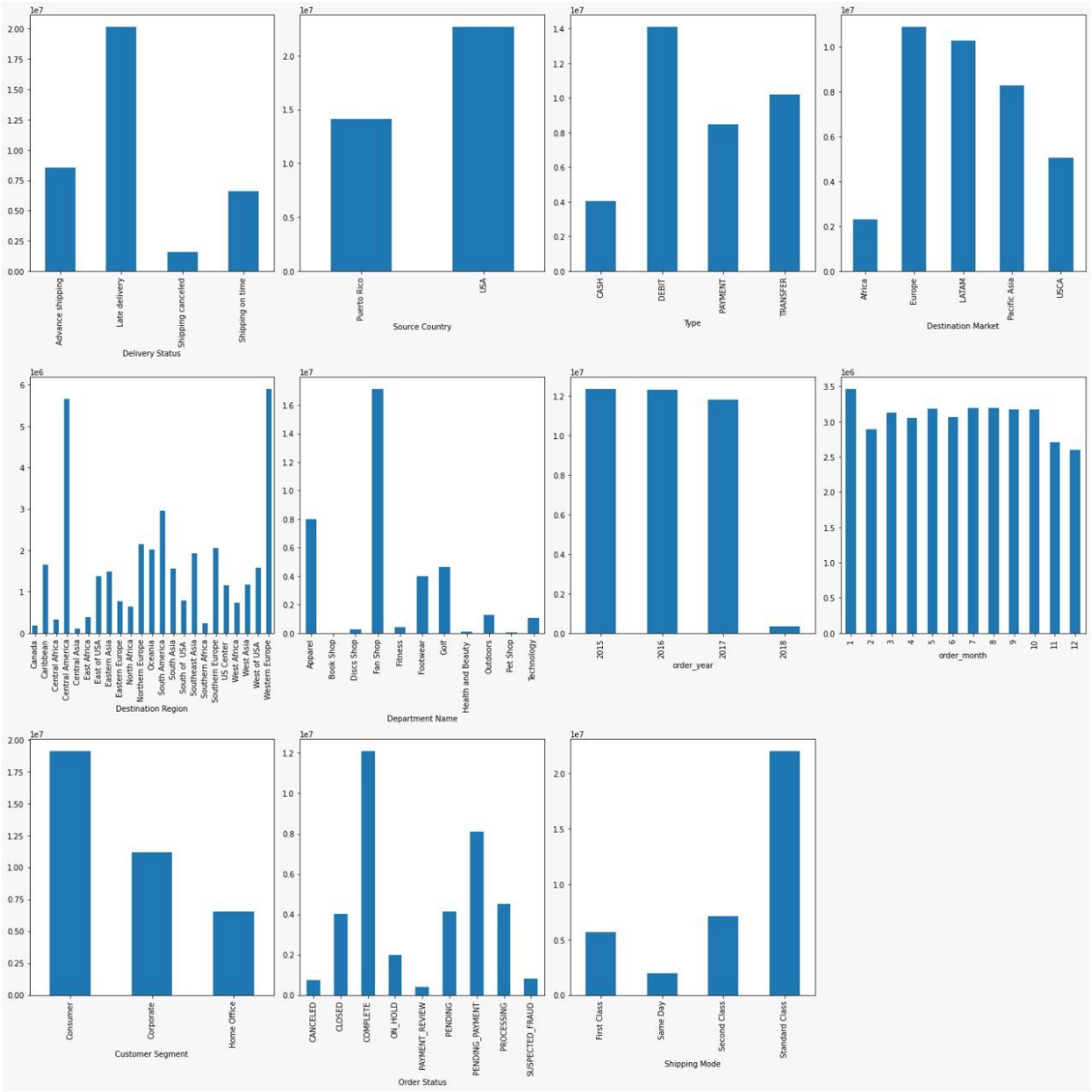
RELATIONSHIP WITH TARGET VARIABLE (Sales):

Numerical independent variables vs target variable(Sales):



- From the above analysis we can see that the variables order item product price,sales per customer,product price has relation with the target variable sales and are dependent.

Categorical independent variables vs target variable(Sales):



- From the Delivery Status we can observe that though the deliveries are late the sales are high.
- The USA has the highest Sales than Puerto Rico.
- Most of the customers chose Debit card as their payment mode.
- Europe has the highest sales followed by LATAM and pacific asia and Africa has the least sales.

- The regions Western Europe and Central America have the highest sales and central asia,Canada,Southern Africa has the least sales.
- Most of the sales are done in the department of the fan shop.
- The sales are very low in the year 2019.
- January has the highest sales.
- From the Customer Segment we can say that consumers are placing many orders hence the sales are high under the consumers segment.
- The sales are high under standard class shipping mode.That is we can say that most of the customers have chosen standard class as their shipping mode.

STATISTICAL TESTS TO VALIDATE THE OBSERVATION MADE FROM EDA FOR TARGET VARIABLE SALES

So far we have observed that the product price,sales per customer has a significant relationship with sales. We have made these observations by studying the above graphs .

We can further validate these observations using statistical tests like Anova,two sample t test.

Two sample t-test:

The two sample t-test is performed to know whether or not the mean weight between two different variables is equal.

The hypothesis for two sample t-test

H0**:** The two population means are equal**.**
H1**:** The two population means are not equal.

```
for i in df2_num.columns:
    print('pvalue for',i,'is :',(ttest_ind(df2['Sales'],df2[i])).pvalue)
```

```
pvalue for Days for shipping (real) is : 0.0
pvalue for Days for shipment (scheduled) is : 0.0
pvalue for Benefit per order is : 0.0
pvalue for Sales per customer is : 0.0
pvalue for Order Item Discount is : 0.0
pvalue for Order Item Discount Rate is : 0.0
pvalue for Order Item Product Price is : 0.0
pvalue for Order Item Profit Ratio is : 0.0
pvalue for Order Item Quantity is : 0.0
pvalue for Product Price is : 0.0
pvalue for Sales is : 1.0
```

- Since the p-value of all continuous variables is less than 0.05 we reject the null hypothesis and conclude that the two population means are not equal.Therefore, we can say that all the continuous variables have some impact on the target variable (Sales).

Kruskal Wallis test

It is used to check the equality of population medians for more than two independent samples. It is a non-parametric test that works similar to ANOVA, but doesn't require the tests of normality and equal variances to be satisfied.

The null and alternative hypothesis is given as:

Ho:The median of all treatments are the same.

H1: At least one treatment has a different median.

To understand the impact of the Categorical Variables on Sales we conducted a Kruskal Wallis test. If the result of the test indicates dependency (reject the null hypothesis) we can state that there is some impact on the dependent variable.

	Independent variables	Pvalue
0	Type	0.055302
1	Delivery Status	0.000000
2	Source Country	0.691342
3	Customer Segment	0.144866
4	Department Name	0.000000
5	Market	0.000000
6	Order Status	0.000000
7	Shipping Mode	0.073130

From the above statistical results we can interpret that the p value of variables Delivery Status,Department Name,Market,Order Status has less than 0.05.So we reject the null hypothesis and conclude that there is some impact on the dependent variable.

BASE MODEL

BASE MODEL CLASSIFICATION:

Here our target variable is Late Delivery Prediction

We used K-Nearest Neighbor(KNN) Algorithm as base model

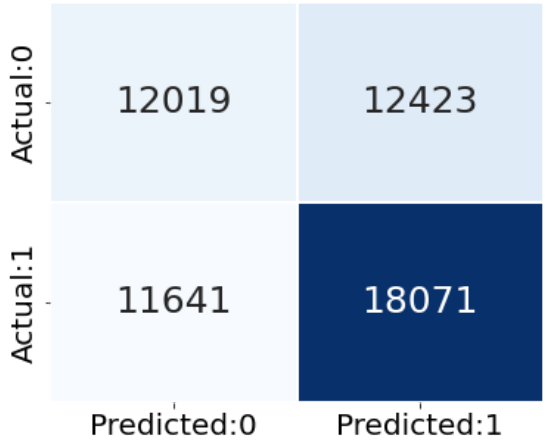
- K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
- KNN is a classification machine learning algorithm used to identify the class of the observation. This algorithm searches for K nearest points to determine the class of an observation. To identify the nearest points, it considers the distance metrics like Euclidean,etc.

```
KNN=KNeighborsClassifier()
model2=KNN.fit(xtrain,ytrain)
ypred2=model2.predict(xtest)
print(classification_report(ytest,ypred2))
```

```
              precision    recall  f1-score   support

           0       0.51      0.49      0.50     24442
           1       0.59      0.61      0.60     29712

    accuracy                           0.56     54154
   macro avg       0.55      0.55      0.55     54154
weighted avg       0.55      0.56      0.55     54154
```



PERFORMANCE EVALUATION METRICS:

- **Confusion Matrix:**
  - ➢ It is the performance measure for the classification problem. It is a table used to compare predicted and actual values of the target variable.
- **Accuracy:**
  - ➢ Accuracy is the fraction of predictions that our model got correct. Higher the accuracy of the model the better is the model. The accuracy of the base model was found to be 0.56
- **Precision:**
  - ➢ Precision is the proportion of positive cases that were correctly predicted.
- **Recall:**
  - ➢ A recall is the proportion of actual positive cases that were correctly predicted.
- **F1 score:**
  - ➢ F1score is the harmonic mean of precision and recall values for a classification model.

BASE MODEL REGRESSION:

- We used Linear Regression as our base model.
- Linear regression is commonly used for predictive analysis.
- Linear regression is used to predict the relationship between two variables ie. independent variable and Target variable.
- It is considered to be significant in business models.

	Metrics	Values
0	Model_name	Base model
1	R-square	0.847332
2	Adjusted Rsquare	0.847213
3	RMSE train	51.824868
4	RMSE test	50.556136
5	MAPE train	0.154968
6	MAPE test	0.158016

PERFORMANCE EVALUATION METRICS:

- **R-Squared:**
  The coefficient of determination explains the percentage of variation in the dependent variable that the independent variables explain collectively.The R-squared of the base model is 0.895.
- **Adjusted R-Squared:**
  It explains the percentage of variation by the independent variables that affect the target variable. The value of adjusted R-squared is always less than or equal to R-squared.The adjusted R-squared of the base model is 0.895.
- **Root Mean Squared Error (RMSE):**
  It is defined as the square root of MSE. Lower the value of RMSE, better is the fit of the regression line.The RMSE for the train data for the base model is found to be 42.79
  The RMSE for the test data for the base model is found to be 41.57.

MODEL BUILDING & METHODS

Feature Engineering:

Scaling:

- As the numerical variables have different units we need to scale the numerical variables to provide equal weightage to all the numerical variables.
- We are using StandardScalar() to Scale the Data.
- This process is a very common pre-processing step.Standard scalar standardizes features of the data set by scaling to unit variance and removing the mean (optionally) using column summary statistics on the samples in the training set

Transforming:

- We observed the skewness of the data and it was not normally distributed.
- To reduce the skewness and make the variables normally distributed we used PowerTransformer()
- One of the assumptions of the regression model is that the residuals should be normally distributed.
- We observed that the skewness of the target variable i.e(sales) was not normally distributed.
- To satisfy the assumption we performed log transformation on the target variable.
- Power transformation is a family of parametric, monotonic transformations that are

applied to make data more Gaussian-like. This is useful for modeling issues related to heteroscedasticity (non-constant variance), or other situations where normality is desired.

Multicollinearity:

- The other assumption of the regression model is that there should be no multicollinearity between the independent variables. We treated it using a variance inflation factor keeping threshold as 5.
- The Variance Inflation Factor (VIF) is used to detect the presence of multicollinearity between the features.
- We calculate VIF of the numerical independent variables.

	Features	Score
0	Days for shipment (scheduled)	4.186913
1	Order Item Discount Rate	2.741503
2	Order Item Product Price	2.396543
3	Order Item Profit Ratio	1.530051
4	Order Item Quantity	2.971159
5	order_month	3.610514

Feature Extraction:

- We extracted the details of Year, Month, Date, Hour from the column Order Date(DateOrdered) into individual columns.

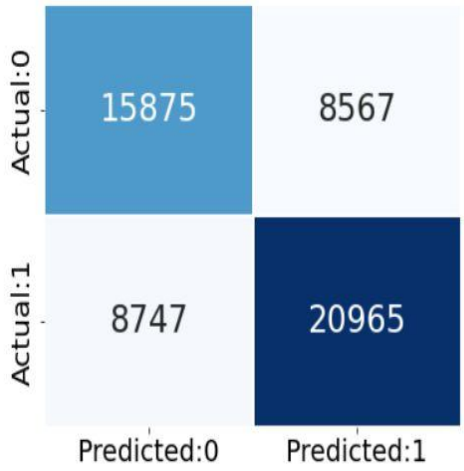
Model Building Classification

KNN Algorithm:

KNN is a classification machine learning algorithm used to identify the class of the observation. This algorithm searches for K nearest points to determine the class of an observation. To identify the nearest points, it considers the distance metrics like Euclidean, Manhattan, Chebyshev, Hamming, and so on.

```
KNN=KNeighborsClassifier()
model1=KNN.fit(xtrain,ytrain)
ypredt=model1.predict(xtrain)
ypred1=model1.predict(xtest)
print(classification_report(ytest,ypred1))
```

	precision	recall	f1-score	support
0	0.64	0.65	0.65	24442
1	0.71	0.71	0.71	29712
accuracy			0.68	54154
macro avg	0.68	0.68	0.68	54154
weighted avg	0.68	0.68	0.68	54154



Logistic regression:

- Logistic Regression is a binary classification algorithm. It predicts the probability of occurrence of a label class.
- Consider that logistic regression is used to identify whether the product falls under the advantage category or not.

```
LR=LogisticRegression()
model2=LR.fit(xtrain,ytrain)
ypredt=model2.predict(xtrain)
ypred2=model2.predict(xtest)
print(classification_report(ytest,ypred2))
```

	precision	recall	f1-score	support
0	0.63	0.90	0.75	24442
1	0.88	0.57	0.69	29712
accuracy			0.72	54154
macro avg	0.76	0.74	0.72	54154
weighted avg	0.77	0.72	0.72	54154

Decision Tree Algorithm:

- Decision trees can be used for classification as well as regression problems.
- The name itself suggests that it uses a flowchart like a tree structure to show the predictions that result from a series of feature-based splits.
- It starts with a root node and ends with a decision made by leaves.

```
DT=DecisionTreeClassifier()
model3=DT.fit(xtrain,ytrain)
ypredt=model3.predict(xtrain)
ypred3=model3.predict(xtest)
print(classification_report(ytest,ypred3))
```

	precision	recall	f1-score	support
0	0.69	0.70	0.70	24442
1	0.75	0.75	0.75	29712
accuracy			0.73	54154
macro avg	0.72	0.72	0.72	54154
weighted avg	0.73	0.73	0.73	54154

Random Forest Algorithm:

- Random Forest consists of several independent decision trees that operate as an ensemble.
- It is an ensemble learning algorithm based on bagging.

```
RF=RandomForestClassifier()
model4=RF.fit(xtrain,ytrain)
ypredt=model4.predict(xtrain)
ypred4=model4.predict(xtest)
print(classification_report(ytest,ypred4))
```

	precision	recall	f1-score	support
0	0.70	0.87	0.78	24442
1	0.87	0.70	0.77	29712
accuracy			0.78	54154
macro avg	0.79	0.78	0.78	54154
weighted avg	0.79	0.78	0.78	54154

HYPER-PARAMETERS

Pre-pruning can be done by specifying the following hyperparameters:

max_depth:

- It is the maximum length of the decision allowed to grow.
- Once the max_depth value is reached the tree will not grow further.

min_samples_split:

- The minimum samples are required to split an internal node.

min_samples_leaf:

- The minimum samples are required to be at the leaf node.
- A node will split further only if its child nodes will have the min_sample_leaf.
- May give the effect of smoothing.

n-estimators:

- This is the number of trees you want to build before taking the maximum voting or averages of predictions.

HYPER-PARAMETER TUNING:

Decision Tree Tuning:

•The Decision tree model is overfitted since there is a significant difference in the training and testing accuracies. Hence hyper tuning the parameters is needed to reduce the overfitting of the model.

•The hyperparameters can be tuned using the Grid Search method. It considers all the combinations of the hyperparameters and returns the optimal hyperparameter values.

•The tuned parameters resulted from the Grid search method as follows:

1. Criterion: Entropy
2. Max_deptth4
3. Min_sample_split:2
4. min_sample_leaf:1
5. max_leaf_nodes:8

```
DT=DecisionTreeClassifier(criterion= 'entropy',max_depth=4,max_leaf_nodes= 8,
                          min_samples_leaf= 1,min_samples_split=2,random_state=10)
model7=DT.fit(xtrain,ytrain)
ypredt=model7.predict(xtrain)
ypred7=model7.predict(xtest)
print(classification_report(ytest,ypred7))
```

```
              precision    recall  f1-score   support

           0       0.64      0.90      0.75     24442
           1       0.88      0.58      0.70     29712

    accuracy                           0.73     54154
   macro avg       0.76      0.74      0.72     54154
weighted avg       0.77      0.73      0.72     54154
```

RandomForest Tuning:

•The Random Forest model is overfitted since there is a significant difference in the training and testing accuracies. Hence hyper tuning the parameters is needed to reduce the overfitting of the model.
•The hyperparameters can be tuned using the Grid Search method. It considers all the combinations of the hyperparameters and returns the optimal hyperparameter values.
•The tuned parameters resulted from the Grid search method as follows:

1. Criterion: Gini
2. Max_depth:10
3. Min_sample_split:2
4. min_sample_leaf:9
5. max_leaf_nodes:11
6. n_estimators:10

```
RF=RandomForestClassifier(criterion='gini',max_depth=10,max_leaf_nodes=11,min_samples_leaf=9,min_samples_split=2,
                          n_estimators=10,random_state=10)
model8=RF.fit(xtrain,ytrain)
ypredt=model8.predict(xtrain)
ypred8=model8.predict(xtest)
print(classification_report(ytest,ypred8))
```

```
              precision    recall  f1-score   support

           0       0.63      0.53      0.57     24442
           1       0.66      0.74      0.70     29712

    accuracy                           0.65     54154
   macro avg       0.64      0.64      0.64     54154
weighted avg       0.64      0.65      0.64     54154
```

Boosting Algorithms:

• To improve the performance of the model, boosting classifiers such as Adaboost, Gradient boosting and XGboost classifiers are considered

Adaboost Algorithm:

• The principle behind boosting algorithms is first we build a model on the training dataset, then a second model is built to rectify the errors present in the first model. This procedure is continued until and unless the errors are minimized, and the dataset is predicted correctly.

```
AD=AdaBoostClassifier()
model5=AD.fit(xtrain,ytrain)
ypredt=model5.predict(xtrain)
ypred5=model5.predict(xtest)
print(classification_report(ytest,ypred5))
```

	precision	recall	f1-score	support
0	0.64	0.89	0.74	24442
1	0.87	0.58	0.70	29712
accuracy			0.72	54154
macro avg	0.75	0.74	0.72	54154
weighted avg	0.77	0.72	0.72	54154

Gradient Boosting Algorithm:

• Ensemble learning involves building a strong model by using a collection (or ensemble) of weaker models. Gradient boosting falls under the category of boosting methods, which iteratively learn from each of the weak learners to build a strong model.

```
GB=GradientBoostingClassifier()
model6=GB.fit(xtrain,ytrain)
ypredt=model6.predict(xtrain)
ypred6=model6.predict(xtest)
print(classification_report(ytest,ypred6))
```

	precision	recall	f1-score	support
0	0.64	0.92	0.76	24442
1	0.90	0.58	0.71	29712
accuracy			0.73	54154
macro avg	0.77	0.75	0.73	54154
weighted avg	0.78	0.73	0.73	54154

XGBoost Algorithm:

XGBoost (extreme gradient boost) is an alternative form of gradient boosting method. This method generally considers the initial prediction as 0.5 and build the decision tree to predict the residuals. It considers the regularization parameter to avoid overfitting.

```
XG=XGBClassifier()
model9=XG.fit(xtrain,ytrain)
ypredt=model9.predict(xtrain)
ypred9=model9.predict(xtest)
print(classification_report(ytest,ypred9))
```

```
              precision    recall  f1-score   support

           0       0.66      0.89      0.76     24442
           1       0.87      0.62      0.72     29712

    accuracy                           0.74     54154
   macro avg       0.76      0.75      0.74     54154
weighted avg       0.77      0.74      0.74     54154
```

Stacking Algorithms:

Stacked generalization consists of stacking the output of an individual estimator and using a classifier to compute the final prediction. Stacking allows us to use the strength of each individual estimator by using their output as input of a final estimator.

```
estimators = [('DT',DecisionTreeClassifier(criterion= 'entropy',max_depth=4,max_leaf_nodes= 8,
                          min_samples_leaf= 1,min_samples_split=2,random_state=10)),
              ('XG', XGBClassifier())]
STA = StackingClassifier(estimators=estimators, final_estimator=XGBClassifier())
model11=STA.fit(xtrain,ytrain)
ypredt=model11.predict(xtrain)
ypred11=model11.predict(xtest)
print(classification_report(ytest,ypred11))
```

```
              precision    recall  f1-score   support

           0       0.66      0.88      0.75     24442
           1       0.87      0.62      0.72     29712

    accuracy                           0.74     54154
   macro avg       0.76      0.75      0.74     54154
weighted avg       0.77      0.74      0.74     54154
```


```
estimators = [('XG',XGBClassifier()),
              ('RF', RandomForestClassifier(criterion='gini',max_depth=10,max_leaf_nodes=11,min_samples_leaf=9,min_samples_split=2,
                          n_estimators=10,random_state=10))]
STA = StackingClassifier(estimators=estimators, final_estimator=LogisticRegression())
model11=STA.fit(xtrain,ytrain)
ypredt=model11.predict(xtrain)
ypred11=model11.predict(xtest)
print(classification_report(ytest,ypred11))
```

```
              precision    recall  f1-score   support

           0       0.66      0.86      0.75     24442
           1       0.85      0.64      0.73     29712

    accuracy                           0.74     54154
   macro avg       0.76      0.75      0.74     54154
weighted avg       0.76      0.74      0.74     54154
```

Score Card:

	desc	accuracy train	accuracy test	precision	recall	f1	Cohen-kappa
KNN	KNN	0.795326	0.680282	0.709908	0.705607	0.707751	0.354872
LogisticReg	LogisticRegression	0.725302	0.722033	0.87904	0.572092	0.693103	0.459113
AD	AdaBoost	0.725587	0.723215	0.869757	0.582795	0.69793	0.460333
GB	GradientBoost	0.736421	0.73411	0.895052	0.583838	0.7067	0.482533
DT_T	DT tuned	0.72793	0.726004	0.876404	0.582795	0.700061	0.466053
RF_T	RF tuned	0.655112	0.64601	0.656326	0.744817	0.697777	0.274708
XG	XGBoost	0.761802	0.74061	0.869894	0.619952	0.723957	0.491996
XGRF	XGBoost RF	0.728689	0.726041	0.87505	0.584074	0.700549	0.465986
STA	Stacking XGB+DT_T	0.761264	0.739262	0.865084	0.621735	0.723495	0.489034
STA_DR	Stacking RF_T+DT_T	0.72793	0.726004	0.876404	0.582795	0.700061	0.466053
STA_XR	Stacking XGB+RF_T	0.769771	0.740075	0.846973	0.642299	0.730572	0.488504

As our Target variable is balanced, we choose Accuracy as our performance metric.

The scorecard shows that the final model, which is a stacking of the XGBoost+ RandomForest Tuned Model with LogisticRegressor as the final estimator, has similar accuracy for the train and test data. As the train and test data give similar results, we can say that our model is a generalized model with no overfitting. The Recall score for this model also looks good, which is important for our business. Hence, we consider this as our final model.

Model Building Regression

Transformed and Scaled Linear Regression model:

Scaling:

- As the numerical variables have different units for different columns we need to scale the numerical variables to provide equal weightage to all the numerical variables.
- We are using StandardScalar() to Scale the Data.

	Metrics	Model2
0	Model_name	Transformation+VIF
1	R-squaretrain	0.888699
2	R-squaretest	0.887607
3	Adjusted Rsquare	0.888667
4	RMSE train	0.228926
5	RMSE test	0.230741

Decision Tree Algorithm:

- Decision trees can be used for classification as well as regression problems.
- The name itself suggests that it uses a flowchart like a tree structure to show the predictions that result from a series of feature-based splits.
- It starts with a root node and ends with a decision made by leaves.

	Metrics	Model3
0	Model_name	DecisionTree
1	R-squaretrain	0.999964
2	R-squaretest	0.992036
3	Adjusted Rsquare	0.999964
4	RMSE train	0.004133
5	RMSE test	0.061421

Stochastic Gradient Descent:

- We use Stochastic Gradient Descent (SGD) which considers a single data point (sample) to perform each iteration.
- Each sample is randomly selected for performing the iteration.

	Metrics	Model4
0	Model_name	SGDRegressor
1	R-squaretrain	0.883673
2	R-squaretest	0.883007
3	Adjusted Rsquare	0.883647
4	RMSE train	0.234038
5	RMSE test	0.235415

Gradient Boosting Algorithm:

- Ensemble learning involves building a strong model by using a collection (or ensemble) of weaker models.
- Gradient boosting falls under the category of boosting methods, which iteratively learn from each of the weak learners to build a strong model.

	Metrics	Model5
0	Model_name	GradientBoosting
1	R-squaretrain	0.995413
2	R-squaretest	0.995581
3	Adjusted Rsquare	0.995412
4	RMSE train	0.046475
5	RMSE test	0.045753

Extreme Gradient Boost(XGboost):

- XGBoost (extreme gradient boost) is an alternative form of gradient boosting method.
- This method generally considers the initial prediction as 0.5 and builds the decision tree to predict the residuals.
- It considers the regularization parameter to avoid overfitting.

	Metrics	Model6
0	Model_name	XGBRegressor
1	R-squaretrain	0.999735
2	R-squaretest	0.994812
3	Adjusted Rsquare	0.999735
4	RMSE train	0.011170
5	RMSE test	0.049574

HYPER-PARAMETER TUNING:

Decision Tree Tuning:

• Hyper tuning the parameters is needed to reduce the overfitting of the model.
•The hyperparameters can be tuned using the Grid Search method. It considers all the combinations of the hyperparameters and returns the optimal hyperparameter values.
•The tuned parameters resulted from the Grid search method as follows:
Best parameters for decision tree Regression: {'criterion': 'squared_error', 'max_depth': 5, 'max_leaf_nodes': 9, 'min_samples_leaf': 1, 'min_samples_split': 2}

	Metrics	Model7
0	Model_name	Tuned decisiontree
1	R-squaretrain	0.890384
2	R-squaretest	0.892185
3	Adjusted Rsquare	0.890360
4	RMSE train	0.227186
5	RMSE test	0.225993

Linear Regression Tuning:

- Hyper tuning the parameters is needed to reduce the overfitting of the model.
- The hyperparameters can be tuned using the Grid Search method. It considers all the combinations of the hyperparameters and returns the optimal hyperparameter values.
- The tuned parameters resulted from the Grid search method as follows:
  Best parameters for decision tree regression: {'n_jobs': -1}

	Metrics	Model8
0	Model_name	Tuned LinearRegression
1	R-squaretrain	0.884309
2	R-squaretest	0.883752
3	Adjusted Rsquare	0.884283
4	RMSE train	0.233397
5	RMSE test	0.234665

Score Card:

Model_name	R-squaretrain	R-squaretest	Adjusted Rsquare	RMSE train	RMSE test
Base model	0.895893	0.899901	0.895862	42.796046	41.578448
Transformation+VIF	0.884297	0.883763	0.884267	0.233409	0.234653
DecisionTree	0.999964	0.992260	0.999964	0.004133	0.060551
SGDRegressor	0.883549	0.883055	0.883525	0.234162	0.235368
GradientBoosting	0.995413	0.995581	0.995412	0.046475	0.045753
XGBRegressor	0.999735	0.994812	0.999735	0.011170	0.049574
Tuned decisiontree	0.890384	0.892185	0.890362	0.227186	0.225993
Tuned LinearRegression	0.884309	0.883752	0.884285	0.233396	0.234665

- We can observe that the R square score for the train and test data is similar for the final model which is Gradient Boosting. As the train and test metrics are similar we can say that the model is not overfit and this shows it is a Stable model.The RMSE score for both train and test is very less.We know that lower the RMSE better is the model .So we are considering this as our best model.

INFERENCES AND RECOMMENDATIONS

Inference: Some of the important features for Classification model to predict if an order placed will be delivered late or not are:

	Features	Importance
274	Shipping Mode_Standard Class	0.4770
271	Order Status_SUSPECTED_FRAUD	0.0961
0	Days for shipment (scheduled)	0.0866
273	Shipping Mode_Second Class	0.0858
272	Shipping Mode_Same Day	0.0603
16	Type_TRANSFER	0.0407

Product Delivery on time is the most critical factor of consumer happiness. Having a well performing operation helps the business meet or surpass the product delivery expectations of customers.

We observed the following features have the most influence on late deliveries based on the final model: ShippingMode,Order Status,Days For Shipment(Scheduled),Type,Destination_country. After deep diving into how these features are influencing the business we have come up with the following recommendations:

- Most of the Late Deliveries are happening through the Second-class and first-class mode,
- In the Order Status,Late Deliveries are mostly occurring through Pending_Status and Processing.
- In Days for shipment,most of the Late Deliveries are happening when the scheduled date is less than or equal to 2 days.
- Most of the Late deliveries are happening when the payment mode is cash.
- For the Source country being the USA ,most of the products are Delivered late.
- Destination Region (Central America and Western Europe) has majority late deliveries

By making changes to these aspects we might decrease the Late Delivery risk and by focusing on above aspects we may be able to ultimately provide a more satisfying experience for the customers.

Some of the important features for regression model to predict sales are:

	Features	Importance
3	Order Item Product Price	0.626200
5	Order Item Quantity	0.370000
22	Department Name_Technology	0.002800
17	Department Name_Footwear	0.000300
23	Market_Europe	0.000200

Cost is the most critical factor of consumer happiness which is affected directly on the business.Having a well performing operation helps the business meet or surpass the product expectations of customers.It is important to retain customers.It is important to keep customers satisfied by providing the product at the affordable price possible.

It has been discovered that both Western Europe and Central America are the regions with the highest number of sales but also the company lost most revenue from these regions only.

Most people prefer to do payment through debit card and all the fraud transactions are happening with wire transfer so the company should be careful when customers are using wire transfer as the company was scammed with more than 100k by a single customer.

CONCLUSION:

- From our business perspective, it is crucial to minimize the two types of false predictions. Being unable to classify a delivery being late has a huge impact on the business as we will never be able to identify areas of improvement. This can ultimately result in the loss of customers. We,therefore, looked at Accuracy as our target metrics.After running different models to improve upon these metrics we found that by using the Stacking of XGboost and RandomForest we were able to create the most efficient model with accuracy 74%.This was an improvement on our base model which had an accuracy 55% .
- After running different models to improve upon R square we found that by using the Gradient Boosting we were able to create the most efficient model with R-square 99%.This was an improvement on our base model which had R-square 89%.

LIMITATIONS, CHALLENGES & SCOPE

Limitations of Data:

Few of the limitations are: -

1. Also, the duration of data collected is from 2015 to 2018. Due to this there isn't even distribution of the data, it does not represent the latest trends & changes in the market.

Challenges:

Few of the challenges faced are: -

1. High cardinality results in huge training effort in model tuning due to increase in model complexity (i.e. more number of features)
2. We also faced challenges on robust model tuning on all the models. Due to computational limitations, we are limited to using GridSearchCV.

Scope:

Scope for some future work is: -

1. Perform hyper parameter tuning for the Boosting model.
2. Exploring Google collab as an option for model training and tuning with faster lead time.
3. Exploring some robust data sampling technique as part of choosing a smaller sample (a true representation of population data) from the population data.