
PREDICTION ANALYSIS ON THE OLIST DATASET

Olist Dataset: Brazilian E-Commerce Dataset

TABLE OF CONTENTS

| | |
|--|------------------------------|
| 1. OBJECTIVE OF THE PROJECT: | 4 |
| 1.1 Problem Statement: | 4 |
| 1.2 Need for A Solution: | 4 |
| 2. BACKGROUND: | 5 |
| 3. DATA COLLECTION PROCESS: | 5 |
| 3.1 Data Type: | 5 |
| 3.2 Data Selection: | 5 |
| 3.3 Data Cleaning: | 5 |
| 4. DATASET DESCRIPTION AND EXPLORATION: | 6 |
| 4.1 Brazilian E-Commerce Public Dataset by Olist: | 6 |
| 4.2 Customers Dataset: | 6 |
| 4.3 Geolocation Dataset: | 7 |
| 4.4 Order Items Dataset: | 9 |
| 4.5 Payments Dataset: | 10 |
| 4.6 Order Reviews Dataset: | 11 |
| 4.7 Order Dataset: | 12 |
| 4.8 Products Dataset: | 13 |
| 4.9 Sellers Dataset: | 14 |
| 5. DIFFICULTIES EXPERIENCED: | 15 |
| 6. ASSUMPTIONS: | 15 |
| 7. EXPLORATORY ANALYSIS: | 16 |
| 8. PREDICTIVE ANALYSIS MODEL: | 22 |
| 8.1 Linear Regression Model: | 22 |
| 8.2 The First Predictive Linear Regression Model: | 22 |
| 8.3 The Second Predictive Linear Regression Model: | 25 |
| 8.4 Logistic Regression Model: | 28 |
| 8.5 Linear Discriminant Analysis Model: | 29 |
| 8.6 Random Forest Model: | 31 |
| 8.7 ROC Curve: | 33 |
| 9. FINDINGS: | 34 |
| 10. RECOMMENDATIONS: | 34 |
| 11. CITATIONS: | 35 |
| 12. APPENDIX | 36 |
| 13. INDIVIDUAL REPORT: | ERROR! BOOKMARK NOT DEFINED. |

TABLE OF FIGURES

| | |
|---|----|
| Figure 1: Brazilian E-Commerce Dataset Overview | 6 |
| Figure 2: Customer Unique Id by Country | 7 |
| Figure 3: Number of orders by state..... | 8 |
| Figure 4: Revenue generated for Olist from each state in Brazil..... | 9 |
| Figure 5: Number of Orders by Payment Type. | 10 |
| Figure 6: Number of Orders by Review score. | 11 |
| Figure 7: Number of Orders by Delivered Month and Review Score. | 12 |
| Figure 8: Number of Orders by State..... | 13 |
| Figure 9: Number of orders and revenue b order delivered month..... | 16 |
| Figure 10: Number of Orders by Product Category. | 17 |
| Figure 11: Top 5 Revenue of seller_id and review_score by seller_id..... | 17 |
| Figure 12: Number of Orders by City. | 18 |
| Figure 13: Number of Orders by Payment Type. | 18 |
| Figure 14: Freight Value by Location. | 19 |
| Figure 15: Number of Orders by Product Name Length. | 19 |
| Figure 16: Number of Orders and Revenue by per photo quantity..... | 20 |
| Figure 17: Number of Orders delivered prior, on and post their estimated delivery time. | 20 |
| Figure 18: Correlation matrix with respect to Review score of the First Linear Regression Model. | 23 |
| Figure 19: Correlation Plot of First Linear Regression Model..... | 24 |
| Figure 20: Accuracy of the First Linear Regression Model..... | 24 |
| Figure 21: Correlation Matrix with respect to review score of the First Linear Regression Model..... | 26 |
| Figure 22: Correlation Plot of the Second Linear Regression Model..... | 27 |
| Figure 23: Confusion Matrix and Overall Statistics of Logistic Regression model. | 29 |
| Figure 24: Confusion Matrix and Overall Statistics of Linear Discriminant Analysis Model. | 30 |
| Figure 25: Confusion Matrix and Overall Statistics of RandomForest model..... | 32 |
| Figure 26: ROC Curve | 33 |

1. Objective of the Project:

To predict the customer review score of service provided by Olist, a Brazilian e-commerce store, and classify the review score into one of the two classes - Poor, Good. This classification is then used to derive actionable insights that assist in making business decisions.

1.1 Problem Statement:

Our problem is to find a way to estimate, based on data about the product and order, what will be the customer review score.

1.2 Need for A Solution:

Olist is a player in the Brazilian e-commerce industry, which is a rapidly evolving field with unpredictable dynamics. Most companies in the E-commerce industry around the world are adapting a customer centric approach to boost their sales. Analysing customer data to check for any existing gap between customer expectation and the service provided by Olist and understanding the reason for the gap can spur Olist's sales.

Feedback is the cornerstone of customer-centricity. Olist collects feedback from its customers in the form of a review score and a review message. Review score is based on a scale of one to five, with one representing lowest customer opinion of the service and five representing the highest customer opinion. Understanding current patterns in the review score and finding a way to predict customer review score, based on data about the product and order, can springboard Olist's revenue.

2. Background:

This dataset was generously provided by Olist, the largest department store in Brazilian marketplaces. Olist connects small businesses from all over Brazil to channels without hassle and with a single contract. Those merchants can sell their products through the Olist Store and ship them directly to the customers using Olist logistics partners.

After a customer purchases the product from Olist Store a seller gets notified to fulfil that order. Once the customer receives the product, or the estimated delivery date is due, the customer gets a satisfaction survey by email where he can give a note for the purchase experience and write down some comments.

3. Data Collection Process:

3.1 Data Type:

- a) Data was available in form of different csv files on Kaggle.

3.2 Data Selection:

- b) Since we wanted to predict review scores at the order level, we augmented the order and order reviews data set.

3.3 Data Cleaning:

- a) There were missing values for categorical and numerical variables in the dataset. We replaced categorical variables with unknown and numerical variables with 0.
- b) We have considered delivery times as the difference between purchase date and delivered date.
The unit of measurement is number of days.
- c) Time was given in ASCII format and we had to convert it into date-time format (GMT+0).

4. Dataset Description and Exploration:

4.1 Brazilian E-Commerce Public Dataset by Olist:

This is a Brazilian ecommerce public dataset of orders made at Olist Store. The dataset has information of 100k orders from 2016 to 2018 made at multiple marketplaces in Brazil. Its features allow viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and finally reviews written by customers. It also includes a geolocation dataset that relates Brazilian zip codes to latitude and longitude coordinates.

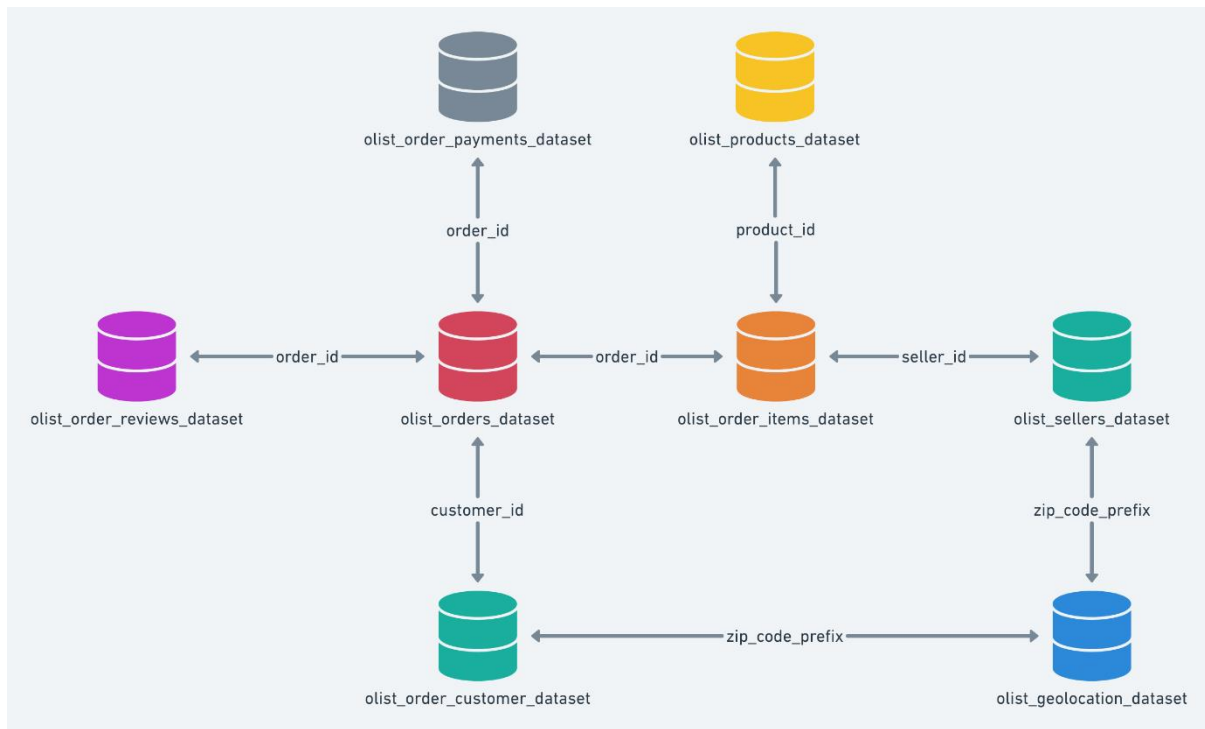


Figure 1: Brazilian E-Commerce Dataset Overview

4.2 Customers Dataset:

This dataset has information about the customer and their location. It is used to identify unique customers in the orders dataset and to find the orders delivery location.

| Variable Name | Description |
|--------------------------|---|
| customer_id | Key to the orders dataset. Each order has a unique customer_id. |
| customer_unique_id | Unique identifier of a customer. |
| customer_zip_code_prefix | First five digits of customer zip code |
| customer_city | Customer city name |
| customer_state | Customer state |

- **Understanding the shape and dimensions of Customers Dataset:** Number of Unique Customer IDs present in the dataset.

Code:

```
str(customers$customer_unique_id)
```

Output:

```
> str(customers$customer_unique_id)
Factor w/ 96096 levels "0000366f3b9a7992bf8c76cfd3221e2"
```



Figure 2: Customer Unique Id by Country

These are a total of 99,441 customers out of which 96,096 are unique users.

4.3 Geolocation Dataset:

This dataset has information Brazilian zip codes and its latitude and longitude coordinates. It is used to plot maps and find distances between sellers and customers.

| Variable Name | Description |
|-----------------------------|----------------------------|
| geolocation_zip_code_prefix | First 5 digits of zip code |
| geolocation_lat | Latitude |
| geolocation_lng | Longitude |
| geolocation_city | City name |
| geolocation_state | State |

➤ **Understanding the shape and dimensions of Geolocation Dataset:**

Code –

```
geolocation <- read.csv("olist_geolocation_dataset.csv",header = T)
str(geolocation)
```

Output –

```
> str(geolocation)
'data.frame': 1000163 obs. of 5 variables
```

1. Number of Orders placed on Olist from each state in Brazil:

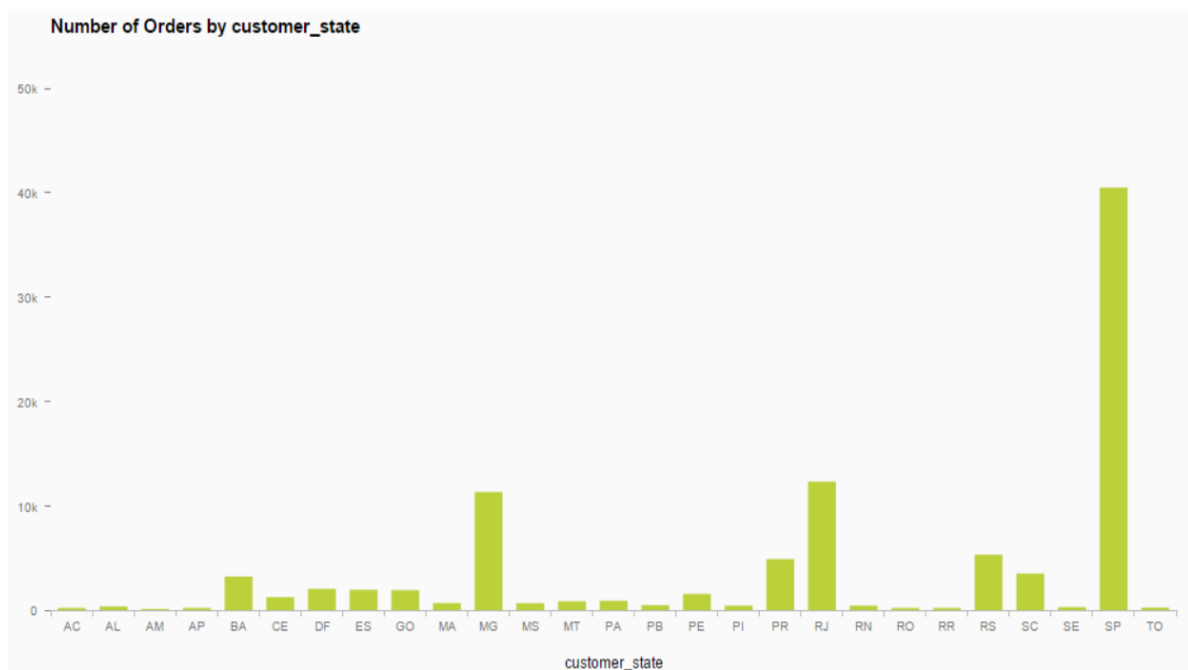


Figure 3: Number of orders by state.

2. Revenue generated for Olist from each state in Brazil:

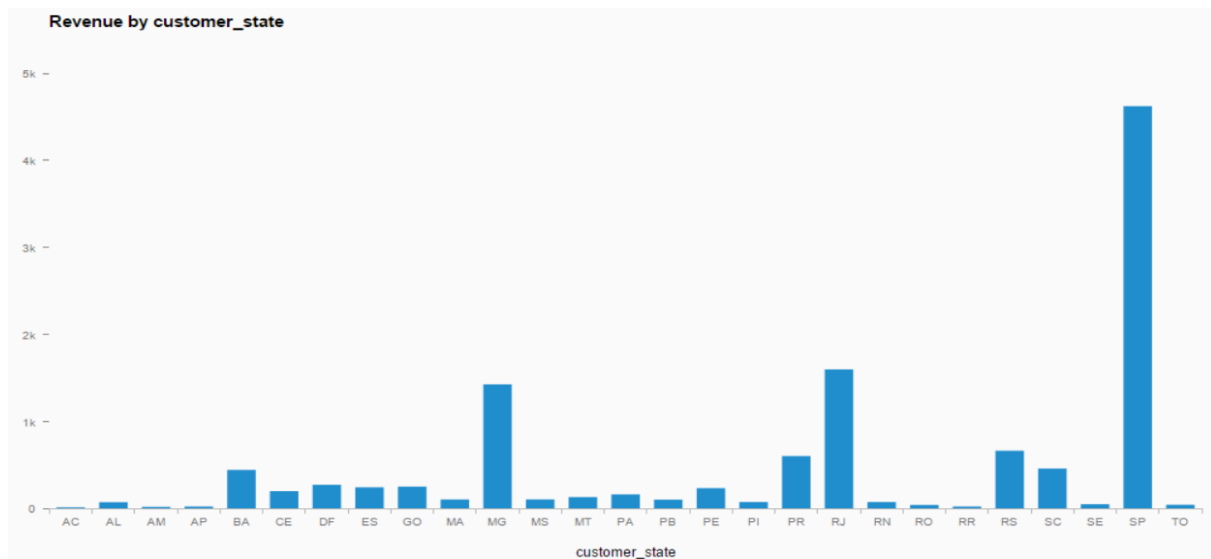


Figure 4: Revenue generated for Olist from each state in Brazil.

4.4 Order Items Dataset:

This dataset includes data about the items purchased within each order.

| Variable Name | Description |
|---------------|--|
| order_id | Unique identifier |
| order_item_id | Sequential number identifying number of items included in the same order. |
| product_id | Product unique identifier |
| seller_id | Seller unique identifier |
| price item | Price |
| freight_value | Item freight value (if an order has more than one item the freight value is split between the items) |

4.2 Understanding the shape and dimensions of Order Items Dataset:

Code –

```
order_items_details <- read.csv("olist_order_items_dataset.csv",header = T)
str(order_items_details)
```

Output –

```
> str(order_items_details)
'data.frame':  112650 obs. of  6 variables
```

4.5 Payments Dataset:

This dataset includes data about the orders payment options. There are four payment methods: credit card, debit card, boleto and voucher. Boleto is a Brazilian payment method which is like a payment/bank slip.

| Variable Name | Description |
|----------------------|---|
| order_id | Unique identifier of an order. |
| payment_sequential | A customer may pay for an order with more than one payment method. If he does so, a sequence will be created to accommodate all payments. |
| payment_type | Method of payment chosen by the customer. |
| payment_installments | Number of instalments chosen by the customer. |
| payment_value | Transaction value. |

4.3 Understanding the shape and dimensions of Payments Dataset:

Code –

```
order_payments <- read.csv("olist_order_payments_dataset.csv",header = T)
str(order_payments)
```

Output –

```
> str(order_payments)
'data.frame': 103886 obs. of 5 variables
```

Number of Orders by Payment Type:

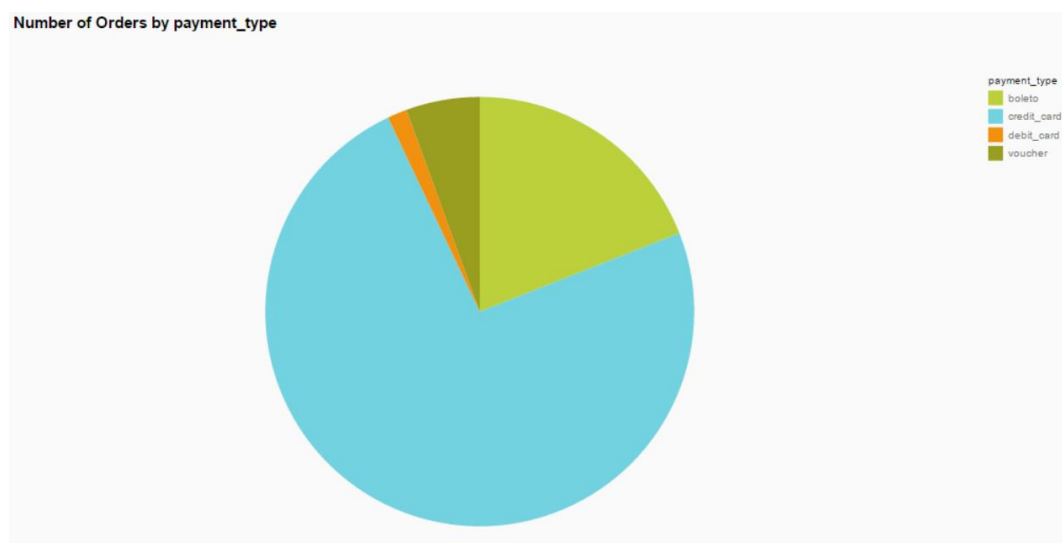


Figure 5: Number of Orders by Payment Type.

Most payments are done with credit card (almost 75%) and another 20% with boleto.

4.6 Order Reviews Dataset:

This dataset includes data about the reviews made by the customers.

| Variable Name | Description |
|-------------------------|---|
| review_id | Unique review identifier |
| order_id | Unique order identifier |
| review_score | Ranging from 1 to 5 and given by the customer on a satisfaction survey. |
| review_comment_title | Comment title from the review left by the customer, in Portuguese. |
| review_comment_message | Comment message from the review left by the customer, in Portuguese. |
| review_creation_date | Shows the date on which the satisfaction survey was sent to the customer. |
| review_answer_timestamp | Shows satisfaction survey answer timestamp. |

➤ **Understanding the shape and dimensions of Order Reviews Dataset:**

Code –

```
order_reviews <- read.csv("olist_order_reviews_dataset.csv",header = T)
str(order_reviews)
```

Output –

```
> str(order_reviews)
'data.frame':  100000 obs. of  7 variables
```

1) Number of Orders by Review Score:

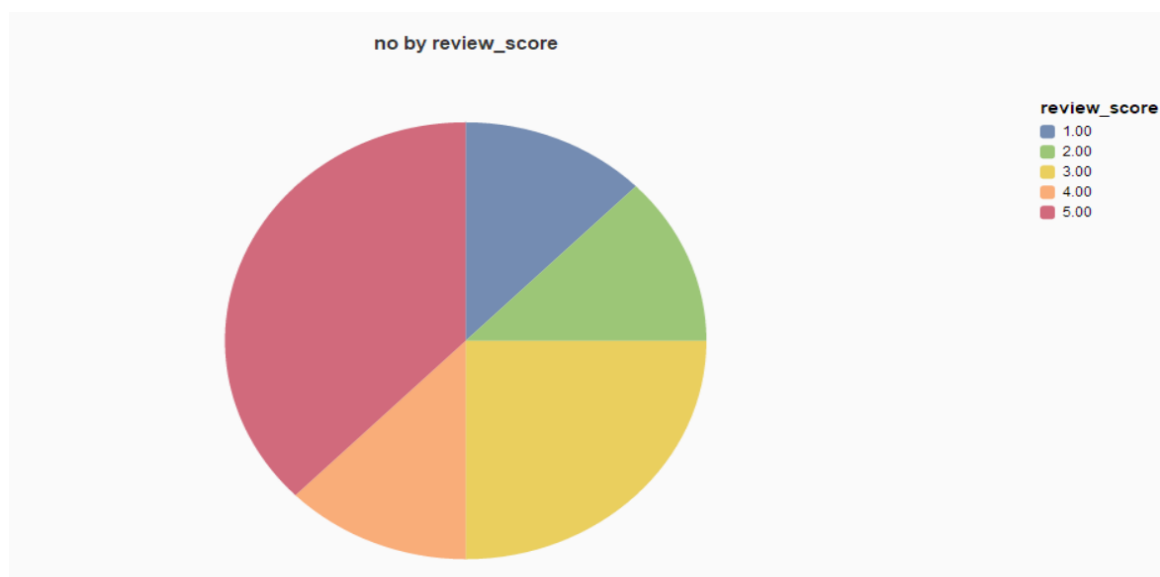


Figure 6: Number of Orders by Review score.

Overall, the products with a review score of 5.0 are ordered the most followed by products with a review score of 3.0.

2) Number of Orders by Delivered Month and Review Score:

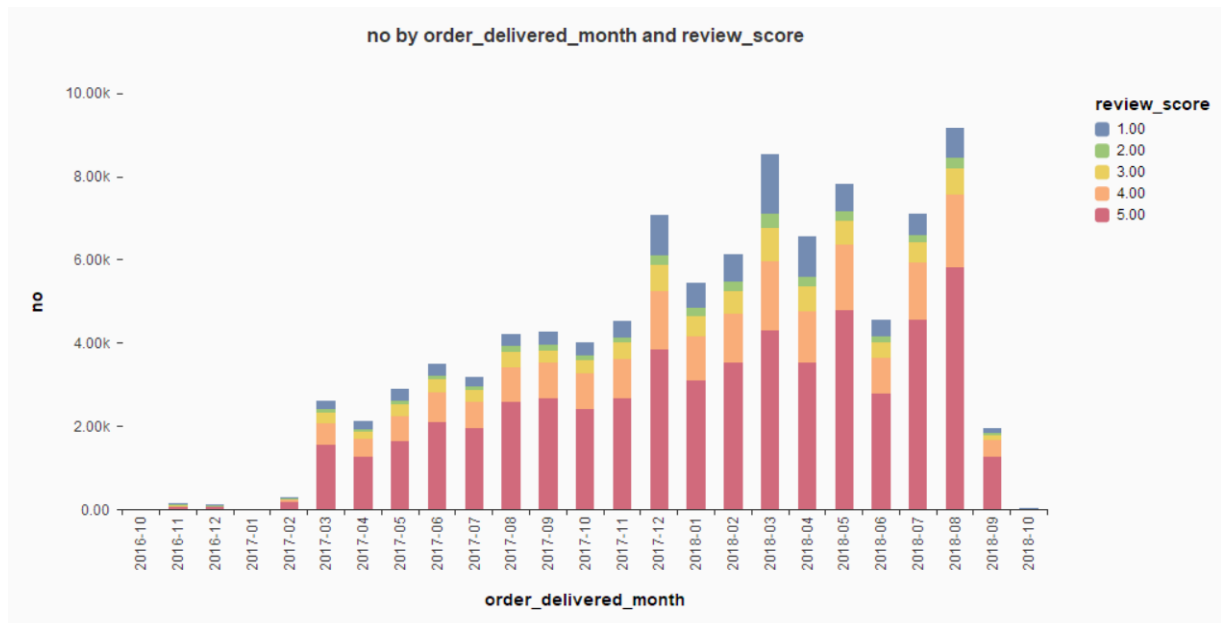


Figure 7: Number of Orders by Delivered Month and Review Score.

The above stacked column chart makes it clear that the proportion of products with a review score of 5.0 being ordered is the highest per any month.

4.7 Order Dataset:

This is the core dataset. From each order we can find all other information.

| Variable Name | Description |
|-------------------------------|---|
| order_id | Unique identifier of the order. |
| customer_id | Key to the customer dataset. Each order has a unique customer_id. |
| order_status | Reference to the order status (delivered, shipped, etc). |
| order_purchase_timestamp | Shows the purchase timestamp. |
| order_approved_at | Shows the payment approval timestamp. |
| order_delivered_carrier_date | Shows the order posting timestamp i.e., when it was handled to the logistic partner. |
| order_delivered_customer_date | Shows the actual order delivery date to the customer. |
| order_estimated_delivery_date | Shows the estimated delivery date that was informed to customer at the purchase moment. |

➤ **Understanding the shape and dimensions of Orders Dataset:**

Code –

```
orders <- read.csv("olist_orders_dataset.csv",header = T,stringsAsFactors = F)
str(orders)
```

Output –

```
> str(orders)
'data.frame':   99441 obs. of  8 variables
```

Number of Orders by State:



Figure 8: Number of Orders by State.

Sao Paulo is also the region with highest number of orders.

4.8 Products Dataset:

This dataset includes data about the products sold by Olist.

| Variable Name | Description |
|----------------------------|--|
| product_id | Unique product identifier |
| product_category_name | Root category of product, in Portuguese. |
| product_name_length | Number of characters extracted from the product name. |
| product_description_length | Number of characters extracted from the product description. |
| product_photos_qty | Number of photos published for the product. |
| product_weight_g | Product's weight measured in grams. |
| product_length_cm | Product's length measured in centimetres. |
| product_height_cm | Product's height measured in centimetres. |

| | |
|------------------|--|
| product_width_cm | Product's width measured in centimetres. |
|------------------|--|

➤ **Understanding the shape and dimensions of Products Dataset:**

Code –

```
products <- read.csv("olist_products_dataset.csv",header = T)
str(products)
```

Output –

```
> str(products)
'data.frame':  32951 obs. of  9 variables
```

4.9 Sellers Dataset:

This dataset includes data about the sellers that fulfilled orders made at Olist. It is used to find the seller location and to identify which seller fulfilled each product.

| Variable Name | Description |
|------------------------|-----------------------------------|
| seller_id | Seller unique identifier |
| seller_zip_code_prefix | First 5 digits of seller zip code |
| seller_city | Seller city name |
| seller_state | Seller state |

➤ **Understanding the shape and dimensions of Sellers Dataset:**

Code –

```
sellers <- read.csv("olist_sellers_dataset.csv",header = T)
str(sellers)
```

Output –

```
> str(sellers)
'data.frame':  3095 obs. of  4 variables
```

5. Difficulties Experienced:

1. The data set was huge. The total file size was 118MB for 9 different *.csv files.
2. There were missing values in variables:

| Variable Name | No. of Missing Values | Variable Name | No. of Missing Values |
|----------------------------|-----------------------|------------------------|-----------------------|
| product_category_name | 610 | product_length_cm | 2 |
| product_name_length | 610 | product_height_cm | 2 |
| product_description_length | 610 | product_width_cm | 2 |
| product_photos_qty | 610 | review_comment_title | 88285 |
| product_weight_g | 2 | review_comment_message | 58245 |

3. As the data set was broken into fragments it needed augmentation using unique identifiers.
4. We had complete data only till August 2018. For the months of September and October we had partial data which affected our analysis.
5. There were different versions of the dataset. Earlier we were planning to use version-2 but then version-6 was released which had more detail on zip codes to increase geolocation precision.

6. Assumptions:

- a) Only considered the orders that were delivered.
- b) We have considered the duration from 4th September 2016 to 17th October 2018.
- c) We have bucketed the review scores as Poor (<3) and Good (>3).

7. Exploratory Analysis:

The dataset has information of 100k orders from 2016 to 2018. It includes 3095 sellers, 71 product categories and 32951 products.

7.1. Number of orders and revenue in thousands by order delivered month.

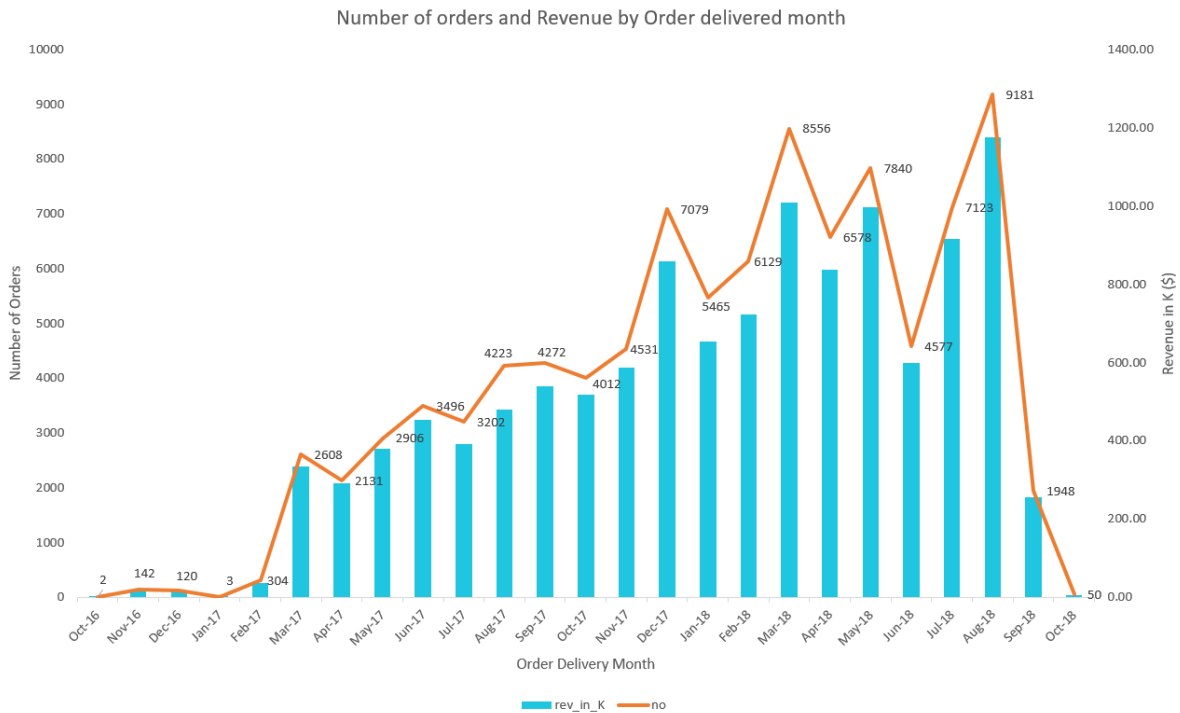


Figure 9: Number of orders and revenue b order delivered month.

From the above graph we can see that the revenue and number of orders have increased over time and that revenue is proportional to the number of orders. August 2018 has the maximum revenue. The percentage increase from 2017 to 2018 over each quarter is as below:

| | 2017 | 2018 | Percentage increase |
|----|---------|---------|---------------------|
| Q1 | 372.12 | 2385.88 | 541.16 |
| Q2 | 1123.77 | 2432.92 | 116.50 |
| Q3 | 1413.45 | 2346.53 | 66.01 |

We can see that there is a considerable increase in revenue from 2017 to 2018. The maximum increase is seen in quarter 1 and the least increase occurs in quarter 3. This can be because the entire data for quarter 3 is not available.

7.2. Top 5 Product categories by number of orders and Top 5 sellers by revenue and their review score



Figure 10: Number of Orders by Product Category.

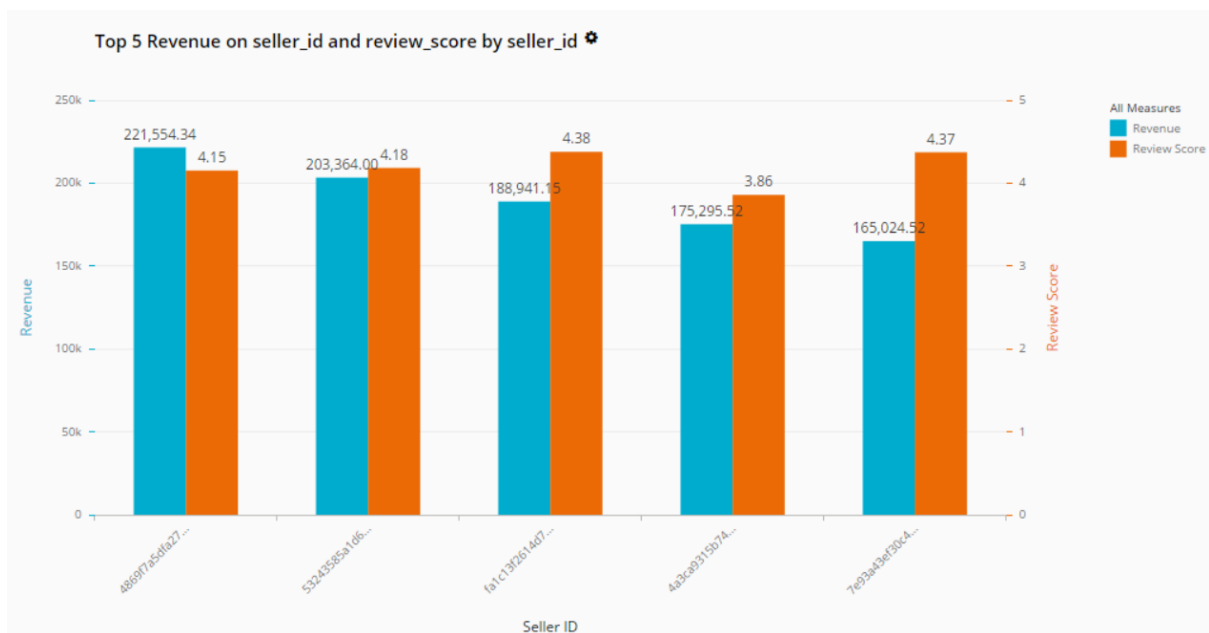


Figure 11: Top 5 Revenue of seller_id and review_score by seller_id.

Cama_mesa_banho translates to Bed-Bath-Table in English, and it is the product category with highest number of orders.

7.3. Number of Orders by City:

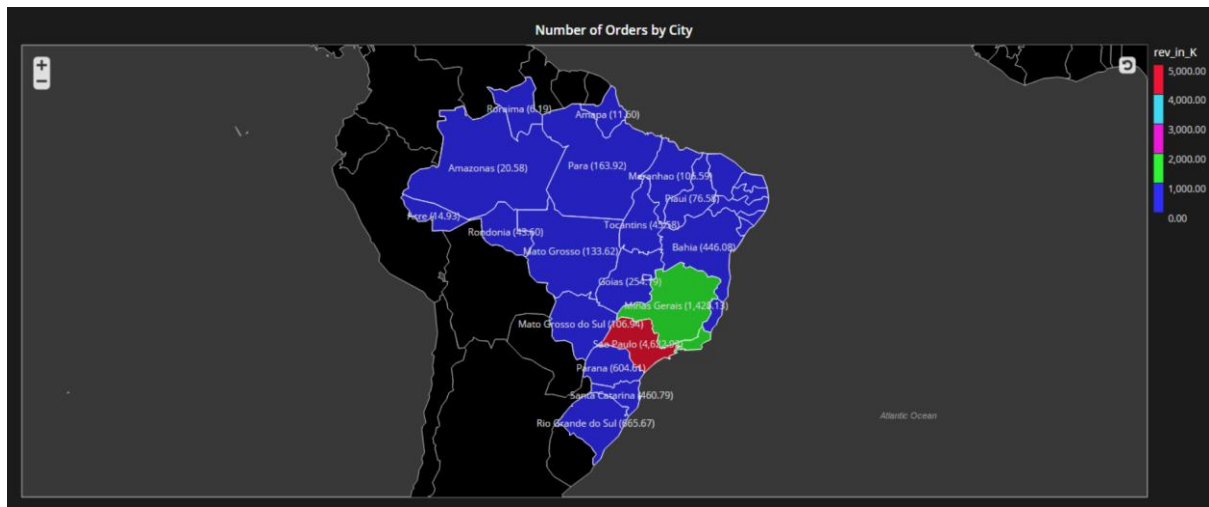


Figure 12: Number of Orders by City.

Sao Paulo has the highest revenue of \$4622.92 thousand and Roraima has the lowest revenue of \$6.19 thousand.

7.4. Number of orders by Payment Type:

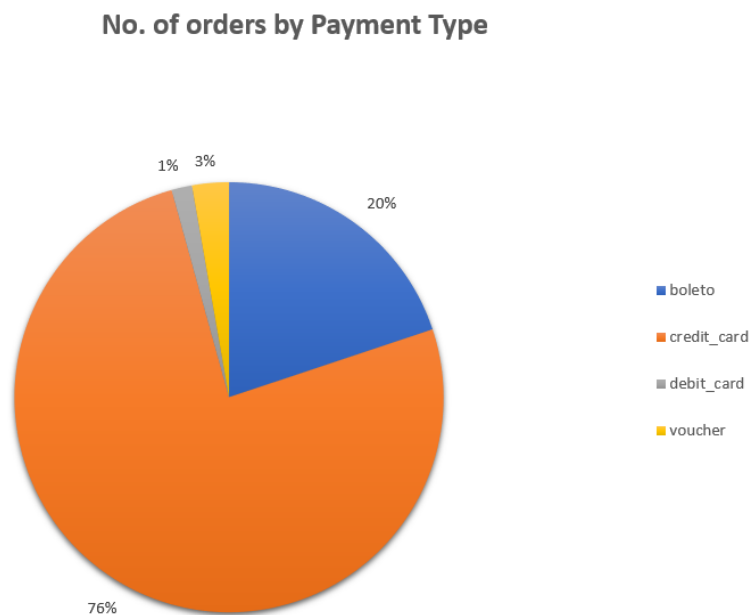


Figure 13: Number of Orders by Payment Type.

From the pie chart we can see that 76% of the orders are paid using credit cards whereas vouchers and debit cards are used only 1% and 3% times respectively.

7.5. Freight value by Location

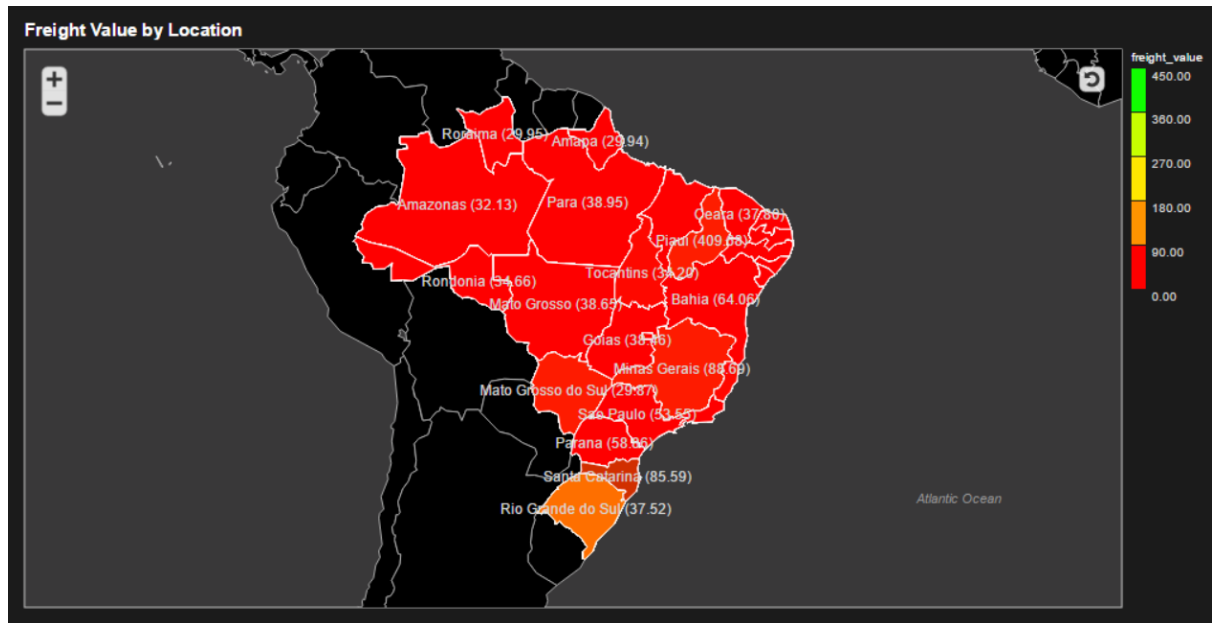


Figure 14: Freight Value by Location.

The above Geo Choropleth chart shows the distribution of freight value among various states in Brazil. Rio Grande do Sul has the highest freight value and hence more hubs should be established there.

7.6. Number of orders by Name Length

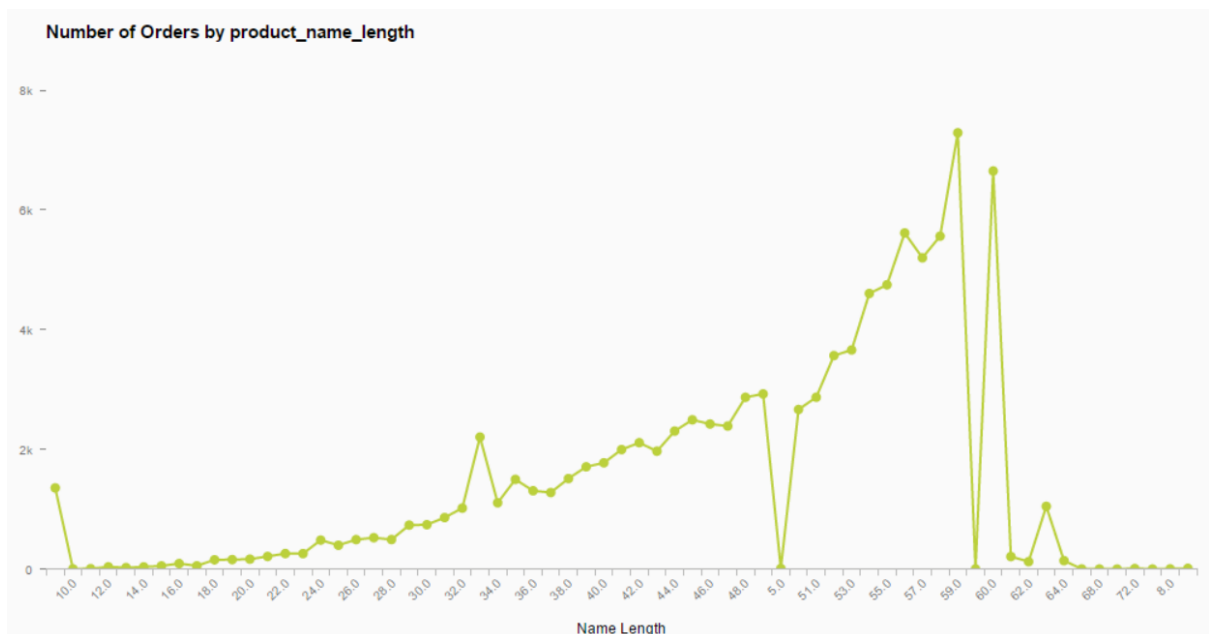


Figure 15: Number of Orders by Product Name Length.

The number of orders increases with increase in the name length. After 60 characters, the number of orders initially falls steeply with increase in the number of characters and then remains consistently low.

7.7. Number of orders and Revenue in thousands per photo quantity:

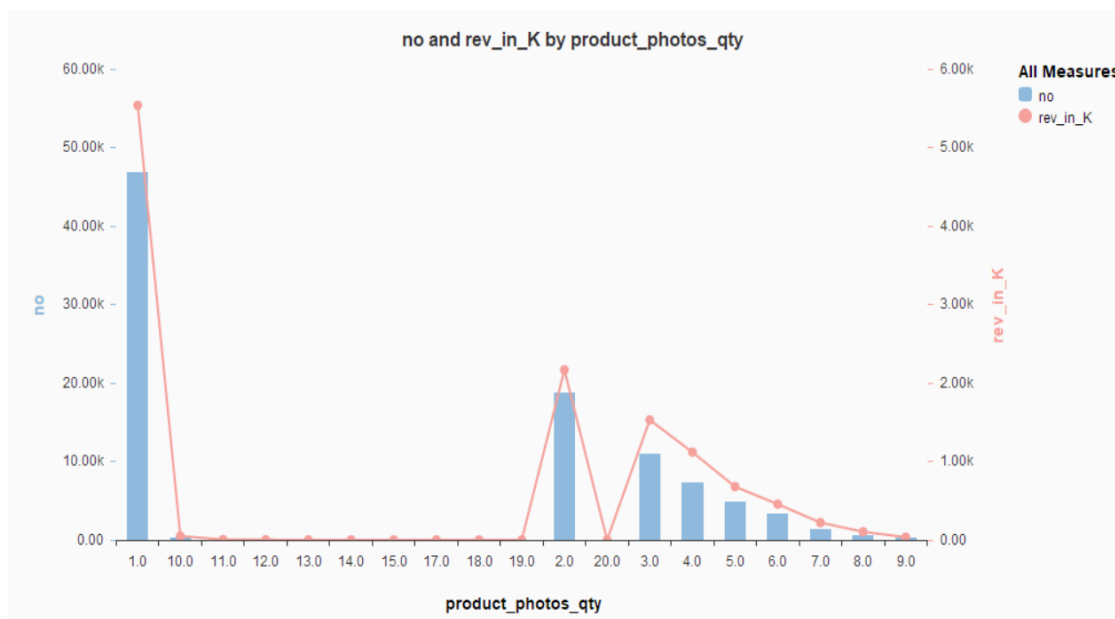


Figure 16: Number of Orders and Revenue by per photo quantity.

The number of orders and revenue are proportional to each other. The products with one photograph listed on Olist have registered highest number of orders and the number of orders decreased as the photo quantity increased. This seems counter intuitive as we expect the products with more photographs listed on the website to receive more orders. One possible explanation for this could be that the products on Olist with only one photograph available are more in quantity than the products with more photographs.

7.8. Number of orders delivered prior, on and post their estimated delivery time:

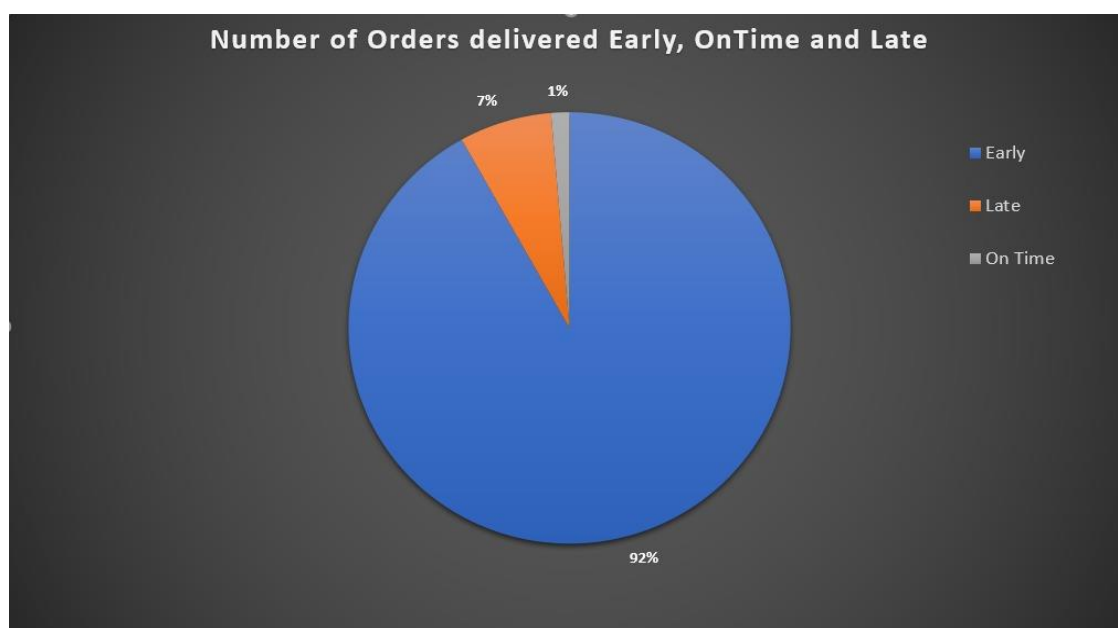


Figure 17: Number of Orders delivered prior, on and post their estimated delivery time.

92% of all orders are delivered before the estimated delivery time, 7% of the orders are delivered late and 1% of all orders are delivered on the estimated delivery day. The below table shows the number of orders for each of these categories.

| Delivery Time | No of Orders |
|----------------------|---------------------|
| Early | 88644 |
| On Time | 1292 |
| Late | 6534 |

8. Predictive Analysis Model:

The objective of the model is to predict the review score of the dataset according to the variables of the dataset and to recommend the variable that matters to the customer the most which will be significant to the review score too. Also, we implement this model to understand logically if the customers churn or not. To implement this objective, we first started with Linear Regression Model.

8.1. Linear Regression Model:

Linear Regression analysis is a basic and the most commonly used type for predictive analysis. Regression is basically used to examine two things:

1. To understand if a set of variables (predictor variables), are doing a good job of determining the prediction outcome, these are dependent variables.
2. To understand which of the predictor variables are significant to predicted outcome variable and in what way do they impact the outcome variable.

Linear Regression refers to a model that can find relationship between two or more variables by fitting a linear equation to observed data. The linear regression line equation: $Y = a + bX$, where X is explanatory variable and Y is the dependent variable.

Linear Regression is a very powerful statistical technique to generate insights on customer behaviour, understanding businesses, factors affecting and influencing profitability, to evaluate trends and make estimates or forecasts. It can also be used to analyse the marketing effectiveness, pricing and sales of the product. It can be used in financial and insurance domains to reduce the risk factor. There is a limited applicability of Linear Regression because it can only work on the dataset which has the dependent variable of continuous nature.

We are generating a linear regression model to predict the review score by analysing the other variables of the dataset and to understand which of the variables are recommended predictors and are significant to the review score variable. Also, the main objective is to generate a good model with good accuracy and the B square value to be at least 95%.

8.2. The First Predictive Linear Regression Model:

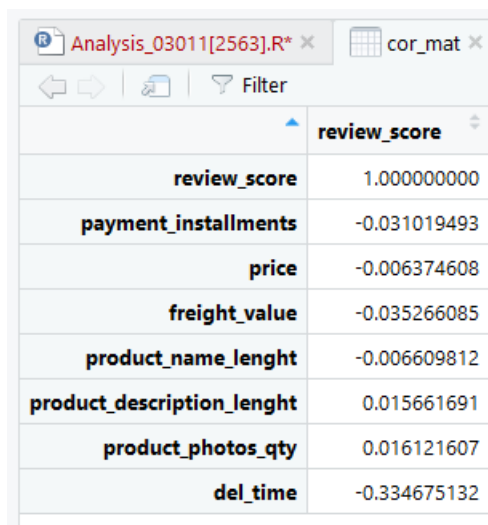
After deciding the objective and the model to use, we started the analysis by first dropping all the variables that are not useful for our prediction. To be specific, the variables which are not good for the prediction of the review score are : status of the order, order delivered carrier date, order purchase timestamp, the time when order was approved, customer zip code prefix, customer unique id, customer city, review id, review comment title, review comment message, review creation date, review answer timestamp, order item id, seller zip code prefix, seller city, product category name, order delivered month, product id, seller id, order id, customer id, order delivered customer date, order estimated delivery date, customer state, seller state, payment sequential, payment value, del time bucket, payment type, product weight g, product height in cm, product length in cm, product width in cm.

We dropped the variables which are not contributing to the predictive analysis so that we get a clear idea behind the variables which might contribute to the predictive analysis, and these variables are then generated into a dataset. Also, to keep the data numerical and clean, we have converted the null values of the dataset to 0 to get better results and accurate model.

After creating the dataset which will contain the variables that will act as the predictor variables for the predictive analysis of review score, we go ahead and create two datasets for our analysis, the training dataset

and validation dataset. The training dataset consists of 80% of the dataset and the validation dataset contains the rest 20%.

After creating the two datasets, we conduct a correlation matrix of the dataset. A correlation matrix table is used to show a correlation coefficient between the variables and each cell in the correlation matrix shows the correlation between two variables. This matrix table is used to summarize the data or as an input for diagnosis for advanced analytics. In our analysis, correlation matrix plays a major role to understand which of the variables have highest correlation with review score. According to the correlation matrix, we find that delivery time has the highest correlation with 33% but in the negative direction. Through this we can state that, if the delivery time increases, the review score for that order is going to be low. We can also understand that the highest correlation is only 33% so there is no strong correlation between the variables with review score and we can assume that this might affect our model.



| | review_score |
|----------------------------|--------------|
| review_score | 1.000000000 |
| payment_installments | -0.031019493 |
| price | -0.006374608 |
| freight_value | -0.035266085 |
| product_name_lenght | -0.006609812 |
| product_description_lenght | 0.015661691 |
| product_photos_qty | 0.016121607 |
| del_time | -0.334675132 |

Figure 18: Correlation matrix with respect to Review score of the First Linear Regression Model.

To understand the correlation with more intensity, we generated a correlation plot, which gives us a graphical structure of the correlation matrix. This gives us an insight and better understanding because when it comes to deciding the strategies and changing the plan of the business, there is not always a programmer in the team. For them to understand this correlation, correlation plot would be the best way to present to them rather than the numerical values. The correlation plot is self-explanatory because it provides a scale which tells us the range of correlation to determine the highest correlated variable.



Figure 19: Correlation Plot of First Linear Regression Model.

Now that we have understood the correlation between the variables and review score, we run a linear regression on the dataset. The summary of the linear regression gives us the knowledge that, delivery time, freight value and product description length play a significant role in predicting the review score. This can be understood because in the real world scenario, if a product's delivery time is very long, the customer is probably not going to give a good review score, if the freight value of a product is almost as much as the product value or maybe a lot more than any other business, it is less likely for a customer to give a good review. Now if the product description length is big and brief describing every detail, the customer is going to like it to get all the information about the product before purchasing it and so might give a good review score. The main objective of the linear regression is to get a high adjusted R squared value, but through this summary we understand that the adjusted R squared value is very low and so the predictors are not a good fit for the model.

To get optimum number of predictors for the model, we do an exhaustive search algorithm. The exhaustive search algorithm is a very general problem-solving technique and algorithmic paradigm to systematically understand and enumerate all possible variables for the model and checks if each of them play a significant role in the model. Running an exhaustive search model gives us the result that there are 7 predictors that play a major role in predicting the review score which are payment installations, price, freight value, product name length, product description length, product photos quantity and delivery time. We run this model on the validation dataset to predict the review score.

By running the linear regression model on the validation dataset and using the exhaustive search model, we get the accuracy of the model. The accuracy of the model gives us the root mean square error (RMSE) value. The RMSE value of the model must be very less for a model to be best fit for predictive analysis, but in our model, we find the RMSE value to be very high, that is, 1.22.

```
> accuracy(a, valid.df$review_score)####RMSE too high,not so good model
              ME      RMSE      MAE      MPE      MAPE
Test set -0.003139183 1.22604 0.9472367 -25.59164 44.8529
> |
```

Figure 20: Accuracy of the First Linear Regression Model.

So, we can now conclude that this model is not best fit for the analysis also adding to the fact that the correlation of direct variables was very less.

8.2.1. Strengths of the Model:

- The summary of the Linear Regression Model gives us the knowledge about the variables that play a significant role in predicting the review score, which are delivery time, freight value and product description length.
- The exhaustive search algorithm gives us the 7 predictors that play a major role in predicting the review score.

8.2.2. Weakness of the model:

- The adjusted R squared value is very low for the model and so the predictors are not a good fit for predicting the review score.
- The RMSE value after the exhaustive search and regression analysis is very high, that is 1.22, so we cannot go ahead with this model.

8.3.The Second Predictive Linear Regression Model:

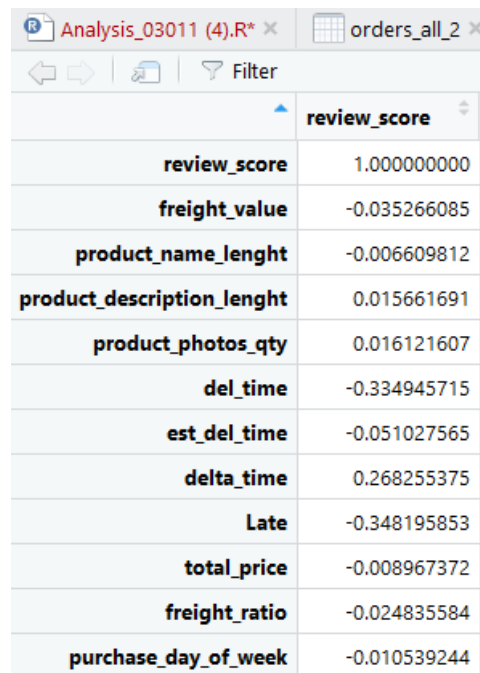
8.3.1. Need for Data Transformation:

To get a better model, we go ahead with an attempt of creating derivative variables, to check if the model works better with the derivative variables or not. The derivative variables are:

- Estimated Delivery Time: To create this derivative variable, we consider the estimated delivery date and the order approved at variable. This derivative variable will give us an insight as to how the customers get affected with the estimated delivery time. It is generally noticed or assumed that if the order approved date and the estimated delivery date are too far away, there might be a decrease in the review score. So, taking that business sense into account, we create this derivative variable.
- Delta Time: The Delta time means the delay in the delivery time, for this variable, we take the difference between the estimated delivery date and the delivered date. If the value is negative, it is assumed that the customers' review score will go down as the order was not delivered at the estimated delivery date and the delivery was delayed. On the same hand, if the value is positive, there is high chances for the review score to be higher.
- Late (Categorical Variable): This variable is a categorical variable consisting of 0 and 1 where the delta time is checked, if the delta time is below zero, that is negative, then the value of Late is 1 which means the delivery has been late. If the delta time is above zero, then the value of Late is 0 which means the delivery has been before or on time. Now through this we can understand that if the delivery has been late, the customers might give bad review scores and if the delivery time is on or before time, the customers might give good review scores.
- Total Price: The total price variable is created by adding the order price of the product and the freight price of the order. This variable gives us the understanding the total price of the order and if the customers must pay more for the product such that it is double of the price of the product or above 30% of the price, then the customers would not be happy and would give a bad review score. This also helps us to understand how the order price is different from the product price because of the freight price.
- Freight Ratio: The freight ratio of the order is defined by the ratio of freight value and the price of the order. This ratio gives us an understanding of how much percent of extra price the customers will have to pay which might affect the revenue of the company. So, if the ratio is higher, the customers will have to pay more and thus might end up giving lower review score. This is a beneficial variable for us to predict the review score.
- Purchase Day of the week: This variable gives us the information regarding which day of the week was the order approved at. It gives a numerical value so that we can consider that for the analysis that we are performing, ranging from 1 to 7. So, this is given by taking order approved at variable taken into consideration.

8.3.2. Running the Linear Regression Model:

Once we have defined the derivative variables, we cumulated the 7 strong predictors and these derivative variables into a dataset for our further analysis. From this analysis, we can derive if there is any improvement in the accuracy by checking the RMSE value and indeed checking if model is better than the previous one. This linear regression model is now implemented by first generating a correlation matrix of the newly created dataset. This correlation matrix will give us an understanding of how correlated the variables are with the review score and if our assumptions of creating the derivative variables are right. So, the correlation matrix gives us the outcome that, the highest correlation is of the Late with review score which 34%.



The screenshot shows the RStudio interface with a script editor titled 'Analysis_03011 (4).R*' and a console window titled 'orders_all_2'. Below the toolbar, a table displays the correlation matrix for the variable 'review_score'. The table has two columns: the variable name and its correlation coefficient. The variables listed are review_score, freight_value, product_name_lenght, product_description_lenght, product_photos_qty, del_time, est_del_time, delta_time, Late, total_price, freight_ratio, and purchase_day_of_week. The correlation values range from 1.000000000 for review_score to -0.010539244 for purchase_day_of_week.

| | review_score |
|----------------------------|--------------|
| review_score | 1.000000000 |
| freight_value | -0.035266085 |
| product_name_lenght | -0.006609812 |
| product_description_lenght | 0.015661691 |
| product_photos_qty | 0.016121607 |
| del_time | -0.334945715 |
| est_del_time | -0.051027565 |
| delta_time | 0.268255375 |
| Late | -0.348195853 |
| total_price | -0.008967372 |
| freight_ratio | -0.024835584 |
| purchase_day_of_week | -0.010539244 |

Figure 21: Correlation Matrix with respect to review score of the First Linear Regression Model.

The correlation matrix gives us the numerical values, but when it comes to comparison, the visual way is preferred and understood by everyone. So, to get the visual view of the matrix, we create the correlation plot which is a graphical representation of the same.

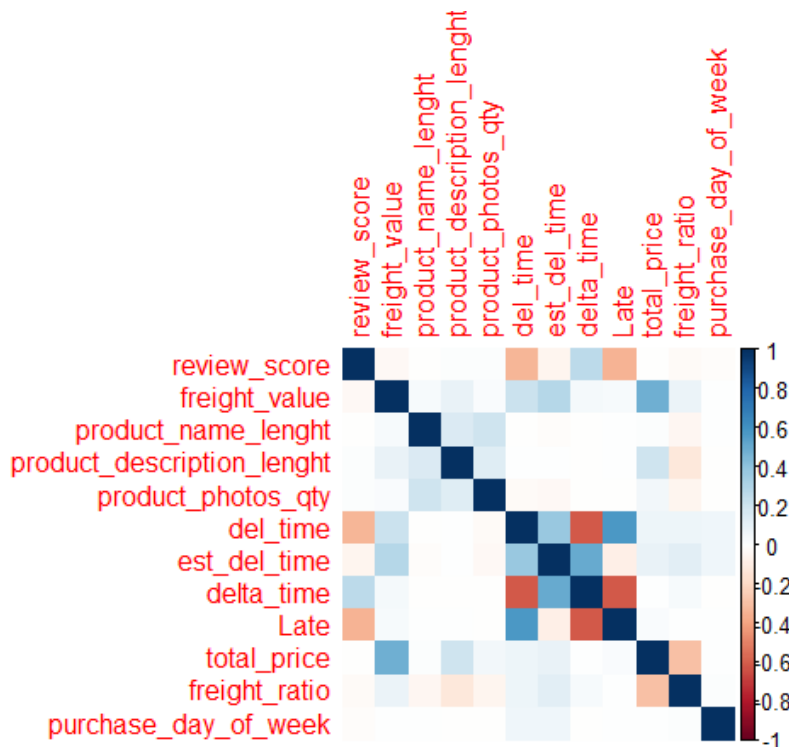


Figure 22: Correlation Plot of the Second Linear Regression Model.

Now from the correlation matrix we also find out that the delta time and late variables which are derivative variables also affect the review score as their correlation value is higher than the rest of the variables. Then we create two datasets again which are the training and validation datasets from this new dataset with 80% and 20% of the dataset respectively. Finally, we run a regression model on this dataset to get the value of the R squared value. This will help us decide if this model is a good fit for our analysis or not. Through this, we understand that, the R squared value, which is 0.15, is better than the previous model but not good enough for the predictive analysis model. After running the linear regression model, we get the RMSE value as 1.20.

```
> accuracy(a_1, valid_1.df$review_score)####RMSE too high,not so good model###
          ME      RMSE      MAE      MPE      MAPE
Test set -0.0007249948 1.20407 0.9282036 -24.32689 43.21304
> |
```

The RMSE value is still very high but it is 1.63% lower than the previous model. So, we do conclude that the dataset created with the 7 strong predictors and derivative variables are better than the previous dataset and can be further used to create better model.

8.3.3. Strengths of the Model:

- Using the dataset including the 7 strong predictors and the derivative variables, we get a better correlation matrix and the best correlation with review score is of LATE with 34%.
- This model gives us a R squared value of 0.15 which is better than the previous model.
- We also get a RMSE value which is 1.63% lower than the previous value which is 1.20.

8.3.4. Weakness of the Model:

- Even though the R squared value is better than the previous model, it is not good enough for the prediction analysis.
- The RMSE value is very high for the analysis model of predicting the review score.

NOTE: We tried the Logistic Regression model with 5 classifications, but we were getting the accuracy of 61% which was not enough for the model, so we decided to make it into 2 classes.

8.4. Logistic Regression Model:

Logistic regression is an instance of classification technique that you can use to predict a qualitative response. It is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

More specifically, you use this set of techniques to model and analyse the relationship between a dependent variable and one or more independent variables. Regression analysis helps you to understand how the typical value of the dependent variable changes when one of the independent variables is adjusted and others are held fixed.

Given X as the explanatory variable and Y as the response variable, how should you then model the relationship between $p(X)=Pr(Y=1|X)$ and X ? The linear regression model represents these probabilities as: $p(X)=\beta_0 + \beta_1X$.

We are using Ordinal Logistic Regression Model; Ordinal regression is used to predict the dependent variable with 'ordered' multiple categories and independent variables. You already see this coming back in the name of this type of logistic regression, since "ordinal" means "order of the categories".

To start off with the logistic regression model, we first had to look into the categories in the review score. We created divided the categories into 2 classes for the logistic regression model. The two classes are: Poor and Good.

8.4.1. The 2 classes: Poor and Good:

- The Poor class consists of the review score which are below the value 3, this means that logically we state the review scores which are below 3 are low and poor and comes under bad category. The Poor category is as 0 in this model.
- The Good class consists of the review score which is greater than 3 and the review score is in a good category if it is above 3 and with a business point of view, getting a review score above 3 should be our objective to retain the customers. Here, the Good category is considered as 1.

8.4.2. Running the Logistic Regression Model:

To run the logistic regression model, we start off by creating the training dataset and the validation dataset, both consisting of 80% and 20% of the dataset respectively. As it is a classification regression model, we need to give the levelling reference for it to give the logistic regression results. So we level the dataset according to the class Poor of the review score as the review score variable has to be predicted. Then we run the logistic regression where it is a multinomial regression and we get the following results.

The Confusion matrix of the logistic regression is shown below as the table and the accuracy of the logistic regression is giving us 88%.

```

> caret::confusionMatrix(cm)
Confusion Matrix and Statistics

      0      1
0    524    175
1    2048   16548

      Accuracy : 0.8848
      95% CI   : (0.8802, 0.8893)
    No Information Rate : 0.8667
    P-Value [Acc > NIR] : 2.56e-14

      Kappa : 0.2793
  Mcnemar's Test P-Value : < 2.2e-16

      Sensitivity : 0.20373
      Specificity : 0.98954
    Pos Pred Value : 0.74964
    Neg Pred Value : 0.88987
      Prevalence : 0.13330
    Detection Rate : 0.02716
    Detection Prevalence : 0.03623
    Balanced Accuracy : 0.59663

      'Positive' Class : 0
> |

```

Figure 23: Confusion Matrix and Overall Statistics of Logistic Regression model.

The sensitivity is defined as the proportion of positive results out of the number of samples which were actually positive, from the confusion matrix we can derive that the sensitivity is 20%. Similarly, when there are no negative results, specificity is not defined. So, from the matrix we can derive that the specificity is 98%.

The positive predictive value is defined as the percent of predicted positives that are actually positive while the negative predictive value is defined as the percent of negative positives that are actually negative. According to the confusion matrix of logistic Regression model, the Positive Predicted Value is 74% and Negative Predicted Value is 88%.

As the accuracy of the model is 88%, it is good enough for us to go ahead and use this model for our prediction, but to reduce the error rate of the model which is 12%, we wanted to try more models and see if we get a better result and better accuracy.

8.4.3. Strengths of the Model:

- We get the accuracy of the model as 88% by creating 2 classes of the review score and using that in the dataset.
- We also get the Positive Predicted value of the review score of is 74% which is high and important as a business perspective to understand how well the model predicts if the order is going to give us a good review score or not.

The next model that we implement is Linear Discriminant Analysis.

8.5. Linear Discriminant Analysis Model:

Linear discriminant analysis (LDA), normal discriminant analysis (NDA), or discriminant function analysis is a generalization of Fisher's linear discriminant, a method used in statistics, pattern recognition and machine

learning to find a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification.

Discriminant analysis is used when groups are known a priori (unlike in cluster analysis). Each case must have a score on one or more quantitative predictor measures, and a score on a group measure. In simple terms, discriminant function analysis is classification - the act of distributing things into groups, classes or categories of the same type.

The MASS package contains functions for performing linear and quadratic discriminant function analysis. Unless prior probabilities are specified, each assumes proportional prior probabilities (i.e., prior probabilities are based on sample sizes). `lda()` prints discriminant functions based on centered (not standardized) variables.

We run the Linear Discriminant Analysis on our dataset to get better results and to understand if it is better for our model.

Then we run the prediction to the LDA and create a confusion matrix to find the accuracy of the model.

After running the confusion matrix, we get the following results.

```
> caret::confusionMatrix(cm_lda)
Confusion Matrix and Statistics

      0      1
0   869   720
1  1703 16003

      Accuracy : 0.8744
      95% CI   : (0.8697, 0.8791)
No Information Rate : 0.8667
P-Value [Acc > NIR] : 0.000765

      Kappa : 0.3517
McNemar's Test P-Value : < 2.2e-16

      Sensitivity : 0.33787
      Specificity : 0.95695
      Pos Pred Value : 0.54688
      Neg Pred Value : 0.90382
      Prevalence : 0.13330
      Detection Rate : 0.04504
      Detection Prevalence : 0.08235
      Balanced Accuracy : 0.64741

      'Positive' Class : 0

> |
```

Figure 24: Confusion Matrix and Overall Statistics of Linear Discriminant Analysis Model.

From this matrix and statistics, we derive that the accuracy of the model is 87%, even though it is good, but we had an accuracy of 88% by using logistic regression model.

We also interpret that the statistics derived by this analysis is not as good as that of the previous model which is the sensitivity and specificity values.

So, looking at this model we can state that the even though this model gives us a good result we will stick to the Logistic Regression model because it gives us a better result than Linear Discriminant Analysis Model.

8.5.1. Strengths of the Model:

- The Linear Discriminant Analysis model gives us an accuracy of 87% which is very good for our analysis.
- The Specificity of the model is 95%, this shows that the model is 95% proficient enough to avoid false alarms, which is good for the prediction analysis of review score.

As this model couldn't give us better result than the Logistic Regression model we would go ahead and implement another model which is the Random Forest model.

8.6. Random Forest Model:

In the random forest approach, many decision trees are created. Every observation is fed into every decision tree. The most common outcome for each observation is used as the final output. A new observation is fed into all the trees and taking a majority vote for each classification model.

Random Forests are like a famous Ensemble technique called Bagging but have a different tweak in it. In Random Forests the idea is to decorrelate the several trees which are generated on the different bootstrapped samples from training Data. And then we simply reduce the Variance in the Trees by averaging them. An error estimate is made for the cases which were not used while building the tree. That is called an OOB (Out-of-bag) error estimate which is mentioned as a percentage. The R package "randomForest" is used to create random forests.

Averaging the Trees helps us to reduce the variance and improve the Performance of Decision Trees on Test Set and eventually avoid Overfitting.

The idea is to build lots of Trees in such a way to make the Correlation between the Trees smaller. Another major difference is that we only consider a Random subset of predictors \sqrt{m} each time we do a split on training examples. Whereas usually in Trees we find all the predictors while doing a split and choose best amongst them. Typically, $m = \sqrt{p}$ where p are the number of predictors.

We run the Random Forest model to understand if this model gives us a better prediction of the review score and if the accuracy of the model is greater than 81%.

8.6.1. Running the Random Forest model:

First the dependent variables are converted into factors as it is important for running the random forest model. So, both the training and validation datasets are converted into factors. Then we run the random forest on the training dataset and predicting the value for accuracy by running it on the validation dataset.

To get the results, we generate the confusion matrix which gives us the confusion matrix table, accuracy and the overall statistics of the model.

So, the confusion matrix is as follows.

```

> caret::confusionMatrix(cm_rfm)
Confusion Matrix and Statistics

pred_rfm      0      1
0      707     254
1     1865    16469

      Accuracy : 0.8902
      95% CI   : (0.8857, 0.8946)
No Information Rate : 0.8667
P-Value [Acc > NIR] : < 2.2e-16

      Kappa : 0.3533
McNemar's Test P-Value : < 2.2e-16

      Sensitivity : 0.27488
      Specificity : 0.98481
      Pos Pred Value : 0.73569
      Neg Pred Value : 0.89828
      Prevalence : 0.13330
      Detection Rate : 0.03664
      Detection Prevalence : 0.04981
      Balanced Accuracy : 0.62985

      'Positive' Class : 0
> |

```

Figure 25: Confusion Matrix and Overall Statistics of RandomForest model.

This model is giving us an accuracy of 89% which is higher than the logistic regression model and so we can say that the best result is obtained from Random Forest Model.

In this model we also interpret that the sensitivity value is 27%, specificity is 98% and the balanced accuracy is 62%.

The error rate of this model is 11% which is less than that of logistic regression model.

8.6.2. Strength of this model:

- The RandomForest model generates a high accuracy which is 89% which is better than any other model used.
- The specificity of the model is very important for prediction analysis and we see that the specificity is 98% which is very high and so it is beneficial for the business of the company. If the company uses this model, there will be high percentage of correct predicted values which is necessary for the success of the company. This will help the company to retain its customers and maintain the level of good review score high.

8.6.3. Weakness of Logistic Regression Model, Linear Discriminant Analysis model and RandomForest Model:

- We compare the confusion matrix of all the three models to give the gist of weakness of some models over the other.
- We find the accuracy of the Linear Discriminant Analysis model is 87% and Logistic Regression model is 88% which is lower than that of RandomForest model which is 89%.
- The Correct Predicted values of 1 for RandomForest model is 16469, for Linear Discriminant Analysis is 16003 and for Linear Regression Model is 16548, so we see that the least is for Linear Discriminant Analysis model.
- The correct predicted values for 0 for RandomForest model is 707, for Linear Discriminant Analysis is 869 and the least is for Linear Regression Model which is 524.

- Comparing the specificity of the three models we get that the highest specificity is of Linear Regression analysis model with 98.5% then the RandomForest model with 98.4% and the last as the Logistic Regression model with 95%.
- The error rate is considered for all the three models and lesser the error rate, better is the model, so keeping that into consideration we have Linear Discriminant Analysis with 13% error rate, Logistic Regression Model with 12% error rate and the least of all RandomForest Model with 11% error rate.

Hence, we conclude that the best model for our prediction analysis is Random Forest model and we will go ahead with this model with an accuracy of 89%.

Thus, to go ahead with this model, we have implemented the ROC curve for this model as follows.

8.7. ROC Curve:

So, when it comes to a classification problem, we can count on an AUC - ROC Curve. When we need to check or visualize the performance of the multi - class classification problem, we use AUC (Area Under the Curve) ROC (Receiver Operating Characteristics) curve. It is one of the most important evaluation metrics for checking any classification model's performance. It is also written as AUROC (Area Under the Receiver Operating Characteristics).

ROC is a probability curve and AUC represent degree or measure of separability. It tells how much model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. By analogy, Higher the AUC, better is the model in distinguishing.

An excellent model has AUC near to the 1 which means it has good measure of separability. A poor model has AUC near to the 0 which means it has worst measure of separability. In fact it means it is reciprocating the result. It is predicting 0s as 1s and 1s as 0s. And when AUC is 0.5, it means model has no class separation capacity whatsoever.

We have generated the ROC by taking X axis as the Specificity, and Y axis as Sensitivity.

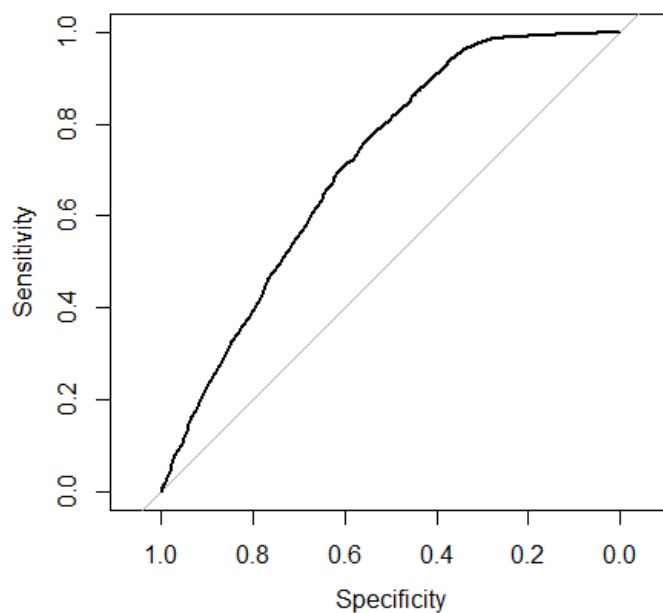


Figure 26: ROC Curve

Through this we derive that the area under the curve is 0.713 which is above 0.5 and is good for separability between the classes and so is good for the prediction model.

9. Findings.

- The RandomForest model generates a high accuracy which is 89% which is better than any other model used.
- The Correct Predicted values of 1 for RandomForest model is 16469, for Linear Discriminant Analysis is 16003 and for Linear Regression Model is 16548, so we see that the least is for Linear Discriminant Analysis model.
- The model depends on the variables:
 - freight_value
 - product_description_length
 - product_photos_qty
 - del_time
 - total_price
 - purchase_day_of_week

10. Recommendations:

To boost the review score, the company can use following recommendations

- They can try to decrease the freight value as it reflects later in the total price. Freight value can be decreased by optimizing shipping that would bring down the transportation cost which would further reduce the freight value.
 - Route optimization can be one of the ways used to reduce the transportation cost
- Product description should be more detailed and lucid as it informs the customer about the product and reduces the perception gap between customer and the product. Thereby, the customer would be satisfied with the product and will give a good review score.
- By increasing the number of photos for the product we can also boost the review score as it would help the customer to get a clear idea of what he is going to get. Thereby limiting any perception bias.
- Delivery time is one of the most important factors affecting the review score. The higher the delivery time the lower the review score. This tell us that the customer is the most dissatisfied with late deliveries. To improve deliveries some methods are
 - Route Optimization
 - Use of Hub and Spoke Model
 - More distribution centres across high demand areas
 - Give discounts to customers for selecting late delivery options
- Total price would always be the most troubling factor in any model. As the customers always wants the most value by paying as little as possible. Therefore, the company could find ways to reduce components of cost like inventory holding or ordering cost. Thereby, reducing the overall price.
- As purchase days of the week also affects the review score. The company could be advised to offer discounts during weekends to boost the review scores of the products.

11. Citations:

Company Website

Statistics Solutions. 2017. "What is Logistic Regression?"
<https://www.statisticssolutions.com/what-is-logistic-regression/>

Company Website

DataCamp. 2018. "Logistic Regression in R Tutorial"
<https://www.datacamp.com/community/tutorials/logistic-regression-R>

Wikipedia

https://en.wikipedia.org/wiki/Linear_discriminant_analysis

website

<https://www.statmethods.net/advstats/discriminant.html>

Blog

<https://www.r-bloggers.com/random-forests-in-r/>

Blog

<https://uberpython.wordpress.com/2012/01/01/precision-recall-sensitivity-and-specificity/>

Medium Article

<https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>

12. APPENDIX

#####Loading Library#####

```
library(dplyr)
library(forecast)
library(leaps)
library(tidyverse)
library(caret)
library(corrplot)
library(nnet)
library(MASS)
library(randomForest)
library(lubridate)
library(pROC)
```

#####Loading datasets#####

```
orders <- read.csv("olist_orders_dataset.csv",header = T,stringsAsFactors = F)
customers <- read.csv("olist_customers_dataset.csv",header = T)
order_reviews <- read.csv("olist_order_reviews_dataset.csv",header = T)
order_payments <- read.csv("olist_order_payments_dataset.csv",header = T)
order_items_details <- read.csv("olist_order_items_dataset.csv",header = T)
sellers <- read.csv("olist_sellers_dataset.csv",header = T)
geolocation <- read.csv("olist_geolocation_dataset.csv",header = T)
products <- read.csv("olist_products_dataset.csv",header = T)
```

#####Data preparation#####

###Merging datasets###

```
orders <- orders[orders$order_status=="delivered",]
orders_all <- merge(orders,customers,by="customer_id",all.x=T)
orders_all$order_purchase_timestamp <-
  as.POSIXct(orders_all$order_purchase_timestamp,"%Y-%m-%d %H:%M:",tz="GMT")
orders_all$order_delivered_customer_date<-
  as.POSIXct(orders_all$order_delivered_customer_date,"%Y-%m-%d %H:%M:",tz="GMT")
orders_all$order_approved_at<-
  as.POSIXct(orders_all$order_approved_at,"%Y-%m-%d %H:%M:",tz="GMT")
orders_all$order_estimated_delivery_date<-
  as.POSIXct(orders_all$order_estimated_delivery_date,"%Y-%m-%d %H:%M:",tz="GMT")
orders_all <- merge(orders_all,order_reviews,by="order_id")
orders_all <- orders_all[!duplicated(orders_all$order_id),]
orders_all <- merge(orders_all,order_payments,by="order_id",all.x = T)
orders_all <- orders_all[!duplicated(orders_all$order_id),]
orders_all <- merge(orders_all,order_items_details,by="order_id",all.X=T)
orders_all <- orders_all[!duplicated(orders_all$order_id),]
orders_all <- merge(orders_all,sellers,by="seller_id",all.X=T)
orders_all <- merge(orders_all,products,by="product_id",all.X=T)
rm(list=setdiff(ls(), "orders_all"))
write.csv(orders_all,file="Orders_merged.csv",row.names = F)
```

#####Loading the combined dataset#####

```
orders_all <- read.csv("Orders_merged.csv",header = T,stringsAsFactors = F)
orders_all$order_purchase_timestamp <-
  as.POSIXct(orders_all$order_purchase_timestamp,"%Y-%m-%d %H:%M:",tz="GMT")
orders_all$order_delivered_customer_date<-
  as.POSIXct(orders_all$order_delivered_customer_date,"%Y-%m-%d %H:%M:",tz="GMT")
```

```
orders_all$order_approved_at<-
  as.POSIXct(orders_all$order_approved_at,"%Y-%m-%d %H:%M:",tz="GMT")
orders_all$order_estimated_delivery_date<-
  as.POSIXct(orders_all$order_estimated_delivery_date,"%Y-%m-%d %H:%M:",tz="GMT")
orders_all$order_delivered_month <-
  format(as.Date(orders_all$order_estimated_delivery_date),"%Y-%m")
orders_all$del_time <- difftime(orders_all$order_delivered_customer_date,
  orders_all$order_purchase_timestamp,
  units="days")
```

```
orders_all$del_time[is.na(orders_all$del_time)] <- 0
orders_all$del_time_bucket <- ifelse(orders_all$del_time < 5,"<5",
  ifelse(orders_all$del_time<10,"5-10",
    ifelse(orders_all$del_time < 20,"10-20",
      ifelse(orders_all$del_time<40,"20-40", ">40"))))
```

####Summarisation and data exploration####

```
no_of_orders_month <- orders_all %>% group_by(order_delivered_month) %>%
  summarise(no=n(),rev_in_K=sum(price)/1000,del_time=mean(del_time))
no_of_orders_location <- orders_all %>% group_by(customer_state) %>%
  summarise(no=n(),rev_in_K=sum(price)/1000,del_time=mean(del_time))
no_of_orders_location_month <- orders_all %>% group_by(customer_state,order_delivered_month) %>%
  summarise(no=n(),rev_in_K=sum(price)/1000,del_time=mean(del_time))
no_of_orders_rs <-orders_all %>% group_by(review_score,order_delivered_month) %>%
  summarise(no=n())
no_of_orders_rs_p <-orders_all %>% group_by(review_score,product_category_name) %>%
  summarise(no=n())
no_of_orders_pt <- orders_all %>% group_by(payment_type) %>%
  summarise(no=n(),rev_in_K=sum(price)/1000,del_time=mean(del_time))
no_of_orders_seller <- orders_all %>% group_by(seller_id) %>%
  summarise(no=n(),rev_in_K=sum(price)/1000,del_time=mean(del_time))
no_of_orders_seller <- no_of_orders_seller[order(-no_of_orders_seller$rev_in_K),]
no_of_orders_seller_1 <- no_of_orders_seller[order(-no_of_orders_seller$no),]
no_of_orders_photos <- orders_all %>% group_by(product_photos_qty) %>%
  summarise(no=n(),rev_in_K=sum(price)/1000,del_time=mean(del_time))
no_of_orders_pcm <- orders_all %>% group_by(product_category_name) %>%
  summarise(no=n(),rev_in_K=sum(price)/1000,del_time=mean(del_time))
no_of_orders_nl <- orders_all %>% group_by(product_name_lenght) %>%
  summarise(no=n(),rev_in_K=sum(price)/1000,del_time=mean(del_time))
```

####Data preparation for prediction####

###Selecting variables and dropping columns###

```
drops <- c("order_status","order_delivered_carrier_date","order_purchase_timestamp",
  "order_approved_at","customer_zip_code_prefix","customer_unique_id",
  "customer_city","review_id","review_comment_title","review_comment_message",
  "review_creation_date","review_answer_timestamp","order_item_id","seller_zip_code_prefix",
  "seller_city","product_category_name","order_delivered_month","product_id","seller_id",
  "order_id","customer_id","order_delivered_customer_date","order_estimated_delivery_date",

  "customer_state","seller_state","payment_sequential","payment_value","del_time_bucket","payment_type",
  "product_weight_g",
  "product_height_cm","product_length_cm","product_width_cm","payment_installments")
orders_all_1 <- orders_all[, !(names(orders_all) %in% drops)]
```

#####Creating partition sets#####

```
orders_all_1[is.na(orders_all_1)] <- 0
orders_all_1$del_time <- as.numeric(orders_all_1$del_time)

set.seed(13)
train.index <- createDataPartition(orders_all_1$review_score, p = 0.8, list = FALSE)
train.df <- orders_all_1[train.index, ]
valid.df <- orders_all_1[-train.index, ]
orders_all_1[is.na(orders_all_1)] <- 0
cor_mat <- cor(orders_all_1)
corrplot(cor_mat, method="color")####Red tells us neg correl and blue gives us pos correl ,intensity of
color gives us the strength of correl
```

##We decide that none of the variables are correlated strongly##

#####Linear regression#####

```
lm_1 <- lm(review_score~.,data=train.df)
summary(lm_1)
```

##We didnt use correl matrix because there were categorical variables and correl matrix causes multi colinearity##

#####Reg search#####

```
search <- regsubsets(review_score ~ ., data = train.df, nbest = 1, nvmax = dim(orders_all_1)[2],
  method = "exhaustive")
sum <- summary(search)##Giving us that 7 predictors are the best model##
a <- predict(lm_1,valid.df)####Predicting on the validation dataset##
accuracy(a, valid.df$review_score)#####RMSE too high,not so good model
```

#####since direct variables were of less correlation,we will try some derivative variables and try running to see if there's an improvement #####

```
orders_all_2 <- orders_all
orders_all_2$est_del_time<- difftime(orders_all$order_estimated_delivery_date,
  orders_all$order_approved_at,
  units="days")
orders_all_2$delta_time<- orders_all_2$est_del_time-orders_all_2$del_time
orders_all_2$Late <- ifelse(orders_all_2$delta_time<0,1,0)
orders_all_2$total_price <- orders_all_2$price+orders_all_2$freight_value
orders_all_2$freight_ratio <- orders_all_2$freight_value/orders_all_2$price
orders_all_2$purchase_day_of_week <- wday(orders_all_2$order_approved_at)
orders_all_2 <- orders_all_2[, !(names(orders_all_2) %in% drops)]
orders_all_2$del_time <- as.numeric(orders_all_2$del_time)
orders_all_2$est_del_time <- as.numeric(orders_all_2$est_del_time)
orders_all_2$delta_time <- as.numeric(orders_all_2$delta_time)
orders_all_2[is.na(orders_all_2)] <- 0
orders_all_2 <- orders_all_2[, -which(names(orders_all_2) %in% c("price"))]
cor_mat_2 <- cor(orders_all_2)
corrplot(cor_mat_2, method="color")
```

#####Partition the dataset#####Using the new derived metrics#####

```
set.seed(13)
train.index_1 <- createDataPartition(orders_all_2$review_score, p = 0.8, list = FALSE)
```

```

train_1.df <- orders_all_2[train.index_1, ]
valid_1.df <- orders_all_2[-train.index_1, ]

lm_2 <- lm(review_score~.,data=train_1.df,na.action = na.omit)
summary(lm_2)

a_1 <- (predict(lm_2,valid_1.df))###Predicting on the validation dataset##

accuracy(a_1, valid_1.df$review_score)####RMSE too high,not so good model###

```

#####Low Med High####

```

orders_all_2$review_score_1 <- ifelse(orders_all_2$review_score<3,0,1)
orders_all_2$review_score <- orders_all_2$review_score_1
orders_all_2 <- orders_all_2[,1:12]

```

####Next model####--Logistic model--####

```

set.seed(13)
train.index_1 <- createDataPartition(orders_all_2$review_score, p = 0.8, list = FALSE)
train_1.df <- orders_all_2[train.index_1, ]
valid_1.df <- orders_all_2[-train.index_1, ]

```

```

orders_all_2$review_score <- (as.factor(orders_all_2$review_score))###Leveling by giving reference###
logit.reg <- glm(review_score~., data = train_1.df)
summary(logit.reg)

```

###fitting the model and checking accuracy on the training model using a confusion matrix##

```

logit.reg.pred <- as.data.frame(predict(logit.reg, valid_1.df[, -1], type = "response"))
colnames(logit.reg.pred)[1] <- "p"
logit.reg.pred$class <- ifelse(logit.reg.pred$p>0.5,1,0)
cm <- table(logit.reg.pred$class,valid_1.df$review_score)
caret::confusionMatrix(cm)

```

#####Next model -linear discriminant analysis#####

#Running linear discriminant analysis

```

lda <- lda(review_score~.,data=train_1.df)

```

###Checking model on test dataset##

```

pred_lda <- predict(lda, valid_1.df)

```

###Checking accuracy using a confusion matrix##Accuracy is again found to be 87%##

```

cm_lda <- (table(pred_lda$class, valid_1.df$review_score))
caret::confusionMatrix(cm_lda)

```

####Next model-random forest

##First convert the dependant to factors##

```

valid_1.df$review_score <- as.factor(valid_1.df$review_score)
train_1.df$review_score <- as.factor(train_1.df$review_score)

```

##Running randomForest on training##

```

rfm <- randomForest(review_score~., train_1.df)

```

###Predicting on the validation dataset and checking for accuracy###---90%

```
pred_rfm <- (predict(rfm, valid_1.df))
```

```
pred_rfm_p <- as.data.frame(predict(rfm, valid_1.df, type = "prob"))
```

```
cm_rfm <- (table(pred_rfm, valid_1.df$review_score))
```

```
caret::confusionMatrix(cm_rfm)
```

####Out of all the models random forest gives us the best result so we choose this to explain the model####

###Plotting roc curve for random forest###

```
r <- roc(valid_1.df$review_score, pred_rfm_p$`1`)
```

```
plot.roc(r)
```

```
auc(r)
```

###The area under the curve is found to be 0.7103

Lavanya Botlaguduru Kolhapuri
