Data Analysis

Brazil: Olist Company

Duration of Data: 24 Months

Data Synopsis: Brazilian Ecommerce Public Dataset: Retail datasets of 100K orders placed on Olist spanning between Oct'2016 and Sep'2018 across several states. Information is tracked with price, order status, payment, freight, and user review along with many other parameters.

Outcomes: Order Forecasting, Descriptive Analysis & Exploratory Data Analysis for the data set.

Project By

Kaushik Prasad Dey (DBI001_019)

Program

Data Science and Business Innovation 2021

Institute –

Indian institute of Management Nagpur

Table of Contents

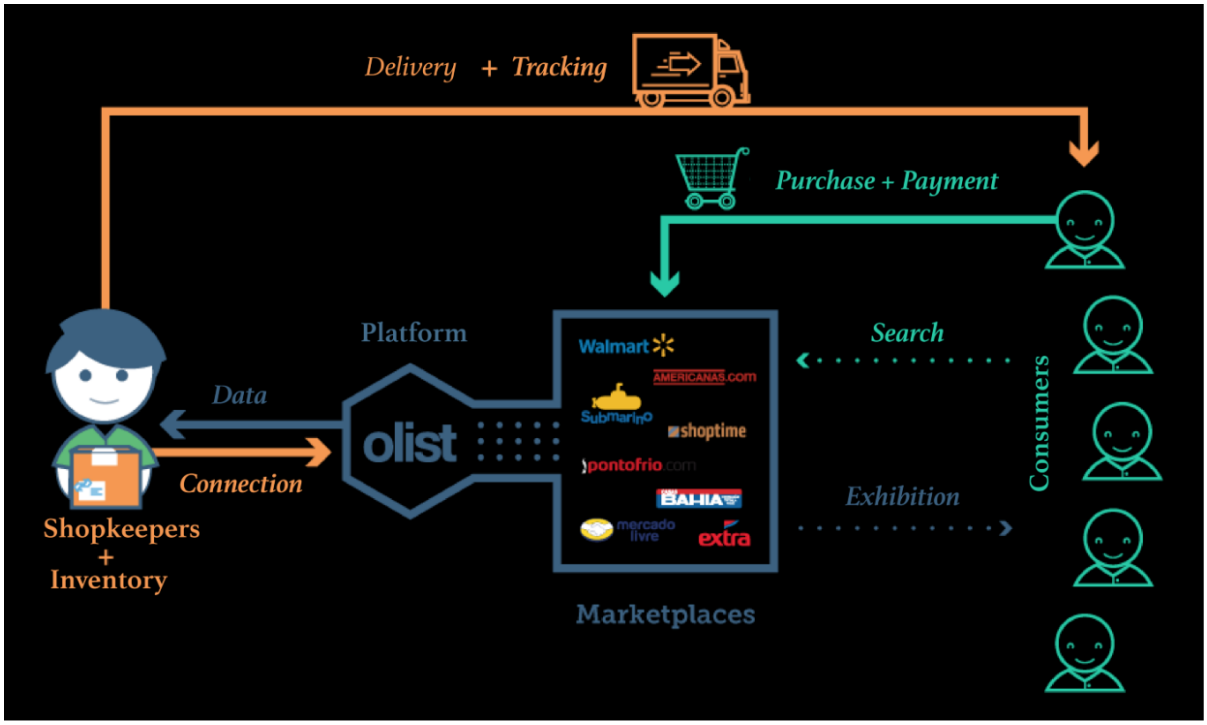
1. Overview of Brazilian Ecommerce company Olist

Olist is Brazilian Online Ecommerce market place for sellers where can get registered and sell products across country. Olist acts as single point of contact between sellers and buyers.

Olist has put ecommerce sales datasets on Kaggle to understand business problems. 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.

Olist connects small businesses from all over Brazil to channels without hassle and with a single contract. Those merchants who can sell their products through the Olist Store and ship them directly to the customers using Olist logistics partners. See more on website: www.olist.com

Olist dataset is available freely on Kaggle



Olist's business model

2. Data Description

Olist dataset have nine sub-datasets which contains which are inter connected via unique columns. Nine datasets are

i) Olist_Customers_Dataset
ii) Olist_Geolocation_Dataset

iii) Olist_Order_Item_Dataset
iv) Olist_Order_Payment_Dataset
v) Olist_Order_Review_Dataset
vi) Olist_Orders_Dataset
vii) Olist_Products_Dataset
viii) Olist_Sellers_Dataset
ix) Olist_Product_Category_Name_Translation

Tools & Libraries used:

3. Data Reading

We have used Python to read and preprocess raw data and tableau for data visualization. We will also use below python libraries for data preprocessing

	dataset	numeric_features	num_features_name	object_features	objt_features_name	bool_features
0	customers	1	customer_zip_code_prefix	4	customer_id, customer_unique_id, customer_city, customer_state	0
1	geolocations	3	zip_code_prefix, geolocation_lat, geolocation_lng	2	geolocation_city, geolocation_state	0
2	items	3	order_item_id, price, freight_value	4	order_id, product_id, seller_id, shipping_limit_date	0
3	payments	3	payment_sequential, payment_installments, payment_value	2	order_id, payment_type	0
4	orders	0		8	order_id, customer_id, order_status, order_purchase_timestamp, order_approved_at, order_delivered_carrier_date, order_delivered_customer_date, order_estimated_delivery_date	0
5	products	7	product_name_lenght, product_description_lenght, product_photos_qty, product_weight_g, product_length_cm, product_height_cm, product_width_cm	2	product_id, product_category_name	0
6	reviews	1	review_score	6	review_id, order_id, review_comment_title, review_comment_message, review_creation_date, review_answer_timestamp	0
7	sellers	1	seller_zip_code_prefix	3	seller_id, seller_city, seller_state	0
8	category_translation	0		2	product_category_name, product_category_name_english	0

- Customer Dataset consist of five columns; Customer_id, customer_unique_id, customer_zip_code_prefix, customer_city, customer_state & 99,441 rows. Out of 5 column's datatypes, **one column is numeric and four are object**

- Geolocation Dataset consists of five columns; zip_code_prefix, geolocation_lat, geolocation_lng, geolocation_city, geolocation_state and 1 million rows. **Out of five column's datatypes, three are numeric and two are object datatypes.**

- Order Item dataset consists of Seven columns; order_id, order_item_id, product_id, seller_id, shipping_limit_date, price, freight_value and 112K rows. **Out of 7 columns, there are 3 numeric datatypes and 4 object datatypes.**

- Payment items consists of five columns; order_id, payment_sequential, payment_type, payment_installments, payment_value and 103K rows. **Out of five columns, there are three numeric and two object datatypes.**

- Order Item datasets consist of eight rows; order_id, customer_id, order_status, order_purchase_timestamp, order_approved_at, order_delivered_carrier_date, order_delivered_customer_date, order_estimated_delivery_date and 99441 rows. **Out of eight columns, all are object datatypes**.

- Product datasets consist of nine columns; product_id, product_category_name, product_name_lenght, product_description_lenght, product_photos_qty, product_weight_g, product_length_cm, product_height_cm, product_width_cm & 32,951 rows. **Out of nine columns, there are seven numeric and two object data**

types.

- Review Dataset consists of seven columns; review_id, order_id, review_score, review_comment_title, review_comment_message, review_creation_date, review_answer_timestamp and 99,224 rows. **Out of seven columns, there is one numeric and six object data types.**

- Seller Dataset consists of 4 columns; seller_id, seller_zip_code_prefix, seller_city, seller_state and 3095 rows. Out of four columns, **one is numeric and three are object data types.**

- Product Category translation consists of two columns; product_category_name, product_category_name_english & 71 rows. All columns are **object data types.**

4. Data Pre-Processing

In order to proceed for descriptive statistics and Exploratory Data Analysis, we need to clean data and remove null values from raw data. We have used a common column id to group all data into a single column.

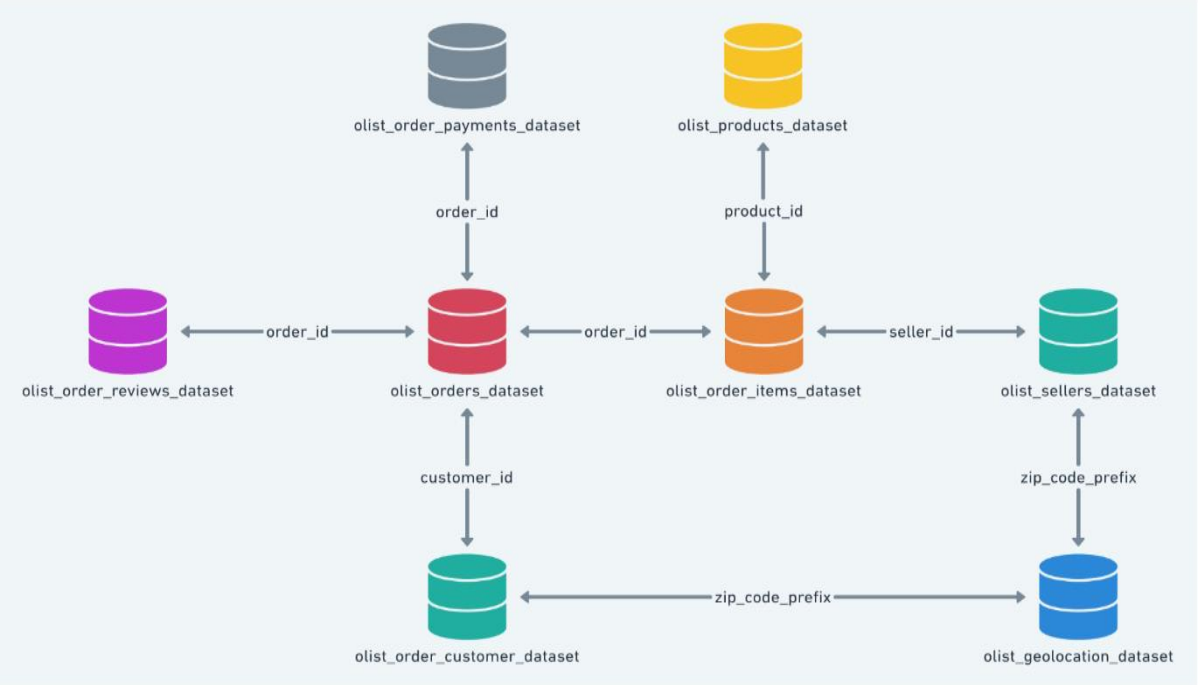
We have used python to wrangle & pre-process data and Tableau for Exploratory Data Analysis.

To Check for No of Rows and Columns in Datasets: Below is the synopsis of work performed

	dataset	no_of_columns	columns_name	no_of_rows
0	customers	5	customer_id, customer_unique_id, customer_zip_code_prefix, customer_city, customer_state	99441
1	geolocations	5	zip_code_prefix, geolocation_lat, geolocation_lng, geolocation_city, geolocation_state	1000163
2	items	7	order_id, order_item_id, product_id, seller_id, shipping_limit_date, price, freight_value	112650
3	payments	5	order_id, payment_sequential, payment_type, payment_installments, payment_value	103886
4	orders	8	order_id, customer_id, order_status, order_purchase_timestamp, order_approved_at, order_delivered_carrier_date, order_delivered_customer_date, order_estimated_delivery_date	99441
5	products	9	product_id, product_category_name, product_name_lenght, product_description_lenght, product_photos_qty, product_weight_g, product_length_cm, product_height_cm, product_width_cm	32951
6	reviews	7	review_id, order_id, review_score, review_comment_title, review_comment_message, review_creation_date, review_answer_timestamp	99224
7	sellers	4	seller_id, seller_zip_code_prefix, seller_city, seller_state	3095
8	category_translation	2	product_category_name, product_category_name_english	71

i. Merge Datasets

We have merged eight datasets in python to proceed for exploratory data analysis. Flow diagram of datasets is shown in below snapshot. We will use common id of datasets to merge datasets.



```
In [8]: df = pd.merge(orders,pay_data, on="order_id")
        df = df.merge(data, on="customer_id")
        df = df.merge(order_itemdata, on="order_id")
        df = df.merge(order_prddata, on="product_id")
        df = df.merge(order_prd_catdata, on="product_category_name")
        df = df.merge(rev_new, on="order_id")
        df = df.merge(order_selldata,on= "seller_id")
```

After merging eight datasets, no of rows are 115,609 & no of columns are 36.

```
 #   Column                         Non-Null Count   Dtype
---  ------                         --------------   -----
 0   order_id                       115609 non-null  object
 1   customer_id                    115609 non-null  object
 2   order_status                   115609 non-null  object
 3   order_purchase_timestamp       115609 non-null  object
 4   order_approved_at              115595 non-null  object
 5   order_delivered_carrier_date   114414 non-null  object
 6   order_delivered_customer_date  113209 non-null  object
 7   order_estimated_delivery_date  115609 non-null  object
 8   payment_sequential             115609 non-null  int64
 9   payment_type                   115609 non-null  object
 29  product_width_cm               115608 non-null  float64
 30  product_category_name_english  115609 non-null  object
 31  review_score                   115609 non-null  int64
 32  review_comment_message         48906 non-null   object
 33  seller_zip_code_prefix         115609 non-null  int64
 34  seller_city                    115609 non-null  object
 35  seller_state                   115609 non-null  object
dtypes: float64(10), int64(6), object(20)
memory usage: 32.6+ MB
```

	order_id	customer_id	order_status	order_purchase_timestamp	order_approved_at	order_delivered_
0	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	2017-10-02 10:56:33	2017-10-02 11:07:15	2017-10
1	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	2017-10-02 10:56:33	2017-10-02 11:07:15	2017-10
2	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	2017-10-02 10:56:33	2017-10-02 11:07:15	2017-10
3	128e10d95713541c87cd1a2e48201934	a20e8105f23924cd00833fd87daa0831	delivered	2017-08-15 18:29:31	2017-08-15 20:05:16	2017-08
4	0e7e841ddf8f8f2de2bad69267ecfbcf	26c7ac168e1433912a51b924fbd34d34	delivered	2017-08-02 18:24:47	2017-08-02 18:43:15	2017-08
...	...	...	...	...	...	
115604	edcf1e1eeb52381be9388c90152be52d	ce2172509c4149d65212484eb761bc37	delivered	2018-08-21 11:29:05	2018-08-21 11:50:47	2018-08
115605	2c12150c742ae2fa48bc703964c16c5f	ab0cf72dfe0538a63a57d6905ccb7b57	delivered	2018-07-28 17:55:27	2018-07-29 18:30:31	2018-07

After merging all datasets, we will check if any null value is present in new dataset. We will use code df.isnull().sum() to check if any null value is there.

It is reported that null values are found with 8 columns; order_approved_at, order_delivered_customer_date, product_weight, product_length_prodcut_height, product_width and review_comment_message have null values.

ii. Dealing with Duplicate data

We need to check data duplication for newly merged datasets. We will create new variable of duplicated rows using pandas. It shows that there are 17,562 duplicate entries in new database. We need to drop these duplicated entries.

3.1.4 Data Dedublicate

```
dup_rows = df[df.duplicated(['order_id','customer_id','order_purchase_timestamp','order_delivered_customer_date','customer_unique
dup_rows.head()
dup_rows.info()
```

3.1.4 Data Dedublicate

In [20]:
```
dup_rows = df[df.duplicated(['order_id','customer_id','order_purchase_timestamp','order_delivered_customer_date','customer_unique
dup_rows.head()
dup_rows.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 17562 entries, 1 to 115608
Data columns (total 35 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   order_id                       17562 non-null  object
 1   customer_id                    17562 non-null  object
 2   order_status                   17562 non-null  object
 3   order_purchase_timestamp       17562 non-null  object
 4   order_approved_at              17562 non-null  object
 5   order_delivered_customer_date  17562 non-null  object
 6   order_estimated_delivery_date  17562 non-null  object
 7   payment_sequential             17562 non-null  int64
 8   payment_type                   17562 non-null  object
 9   payment_installments           17562 non-null  int64
 10  payment_value                  17562 non-null  float64
 11  customer_unique_id             17562 non-null  object
```

We will drop duplicate entries of order id, customer id, order purchase time and order delivered customer date from database.

In [19]:
```
#Deduplication of entries
df= df.drop_duplicates(subset={'order_id','customer_id','order_purchase_timestamp','order_delivered_customer_date'}, keep='first
df=df.reindex()
df
```

Out[19]:

	order_id	customer_id	order_status	order_purchase_timestamp	order_approved_at	order_delivered_cu
0	e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	2017-10-02 10:56:33	2017-10-02 11:07:15	2017-1
3	128e10d95713541c87cd1a2e48201934	a20e8105f23924cd00833fd87daa0831	delivered	2017-08-15 18:29:31	2017-08-15 20:05:16	2017-0
4	0e7e841ddf8f8f2de2bad69267ecfbcf	26c7ac168e1433912a51b924fbd34d34	delivered	2017-08-02 18:24:47	2017-08-02 18:43:15	2017-0
5	bfc39df4f36c3693ff3b63fcbea9e90a	53904ddbea91e1e92b2b3f1d09a7af86	delivered	2017-10-23 23:26:46	2017-10-25 02:14:11	2017-1
6	5f49f31e537f8f1a496454b48edbe34d	a7260a6ccba78544ccfaf43f920b7240	delivered	2017-08-24 11:31:28	2017-08-24 11:45:25	2017-0
...	...	...	...	...	...	
115603	8dcb7601ceb0b144a5fdd0055b91ba28	6b9eb9660bed562d1c735d3fba0cfd60	delivered	2017-06-19 17:11:51	2017-06-19 17:25:18	2017-0

After removing duplicates/Null data, it is reported that number of rows using below syntax:

```
In [20]: print("Number of rows after dedublication:",len(df))
         print("Number of columns after deduplication:",len(df.columns))

         Number of rows after dedublication: 96516
         Number of columns after deduplication: 35
```

All time stamps are in object datatypes, we need to convert them into datetime using below syntax:

```
In [22]: # all time stamps are in object dtype as observed above converting it into dataetime
         df[['order_purchase_timestamp','order_approved_at','order_delivered_customer_date','order_estimated_delivery_date',]]=df[['order_
               'order_approved_at','order_delivered_customer_date','order_estimated_delivery_date']].apply(pd.to_datetime)
```

After eliminating null values, dataset does not have any null value. So, it is ready for exploratory data analysis. Before starting EDA, we have separate date, month, and year columns to provide granular level of details. Below are the overall parameters:

```
In [23]: df.isnull().sum()

customer_zip_code_prefix         0
customer_city                    0
customer_state                   0
order_item_id                    0
product_id                       0
seller_id                        0
shipping_limit_date              0
price                            0
freight_value                    0
product_category_name            0
product_name_lenght              0
product_description_lenght       0
product_photos_qty               0
product_weight_g                 0
product_length_cm                0
product_height_cm                0
product_width_cm                 0
product_category_name_english    0
review_score                     0
review_comment_message           0
```

Now, we need to separate purchase year, purchase month and purchase day from data time. After separating date, month, year, we will get 43 columns and 96,516 rows. (Syntax Used as follows):-

```
In [43]: df.info()
```

```
 26  product_length_cm              96516 non-null  float64
 27  product_height_cm              96516 non-null  float64
 28  product_width_cm               96516 non-null  float64
 29  product_category_name_english  96516 non-null  object
 30  review_score                   96516 non-null  int64
 31  review_comment_message         96516 non-null  object
 32  seller_zip_code_prefix         96516 non-null  int64
 33  seller_city                    96516 non-null  object
 34  seller_state                   96516 non-null  object
 35  order_purchase_year            96516 non-null  int64
 36  order_purchase_month           96516 non-null  int64
 37  order_purchase_month_name      96516 non-null  object
 38  order_purchase_year_month      96516 non-null  object
 39  order_purchase_date            96516 non-null  object
 40  order_purchase_day             96516 non-null  object
 41  order_purchase_hour            96516 non-null  int64
 42  order_purchase_time_day        96516 non-null  category
dtypes: category(1), datetime64[ns](4), float64(10), int64(9), object(19)
memory usage: 31.8+ MB
```

After removing duplicate customer unique ids, there are total 96516 rows and 43 columns. We will use these datasets for exploratory data analysis.

```
In [43]: df = df.drop_duplicates(subset=["customer_unique_id"])
         #now city name should be starting with Capital Letter
         df["customer_city"]= df["customer_city"].str.capitalize()
         df.head()
```

5. Descriptive Analysis of Brazilian Ecommerce Olist Datasets

	count	mean	std	min	25%	50%	75%	max
payment_sequential	96516.0	1.022545e+00	0.247935	1.00	1.000000e+00	1.00	1.000000e+00	27.00
payment_installments	96516.0	2.919858e+00	2.711997	0.00	1.000000e+00	2.00	4.000000e+00	24.00
payment_value	96516.0	1.579244e+02	216.773702	0.01	6.006500e+01	103.19	1.753925e+02	13664.08
customer_zip_code_prefix	96516.0	3.516397e+04	29810.746531	1003.00	1.136875e+04	24422.00	5.901500e+04	99980.00
order_item_id	96516.0	1.017355e+00	0.152817	1.00	1.000000e+00	1.00	1.000000e+00	7.00
price	96516.0	1.257427e+02	189.523484	0.85	4.190000e+01	79.00	1.399000e+02	6735.00
freight_value	96516.0	2.021832e+01	15.932140	0.00	1.330000e+01	16.39	2.126000e+01	409.68
product_name_lenght	96516.0	4.885321e+01	9.992646	5.00	4.200000e+01	52.00	5.700000e+01	76.00
product_description_lenght	96516.0	7.940769e+02	654.561736	4.00	3.500000e+02	608.00	9.950000e+02	3992.00
product_photos_qty	96516.0	2.251233e+00	1.746957	1.00	1.000000e+00	2.00	3.000000e+00	20.00
product_weight_g	96516.0	2.107222e+03	3766.913563	0.00	3.000000e+02	700.00	1.813000e+03	40425.00
product_length_cm	96516.0	3.013767e+01	16.147319	7.00	1.800000e+01	25.00	3.800000e+01	105.00
product_height_cm	96516.0	1.650607e+01	13.347404	2.00	8.000000e+00	13.00	2.000000e+01	105.00
product_width_cm	96516.0	2.306221e+01	11.746188	6.00	1.500000e+01	20.00	3.000000e+01	118.00
review_score	96516.0	4.107412e+00	1.329213	1.00	4.000000e+00	5.00	5.000000e+00	5.00

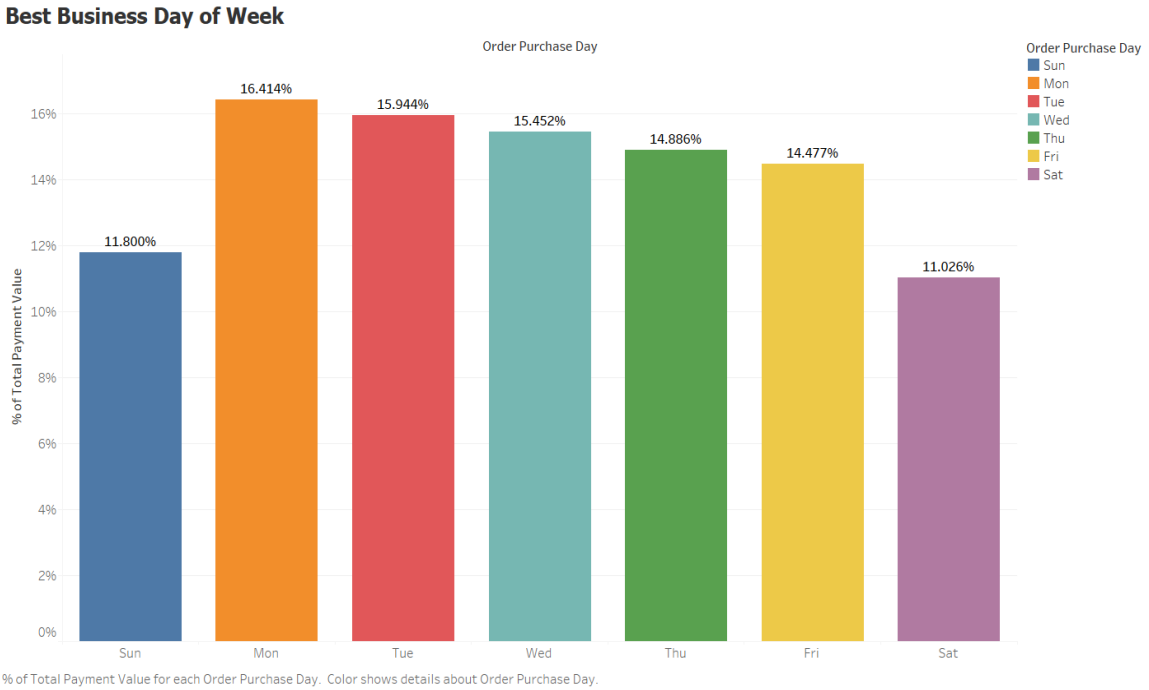
We have performed the analysis defining the correlation between qualitative and quantitative variables. It showcases the Prime values of mean, standard deviation, min value, max value along with other quartiles for all the provided parameters. Based on this we can clearly signify we have normalized all the values so that count of all variables is same. Also, we have identified that max order value has been 13,664 Brazilian real by a single customer. Where average review score has been consistently above 4.

6. Exploratory Data Analysis of Olist Ecommerce Market place

After pre-processing eight datasets and merging them together, we have generated new database which is having 96,516 rows and 43 columns without any missing values. Now, we will be using Tableau software for data visualization.

We will deep dive into data analysis and find useful business insights which will help Olist to keep track of it and focus on core areas to increase customer orders and sales revenues for year 2019.

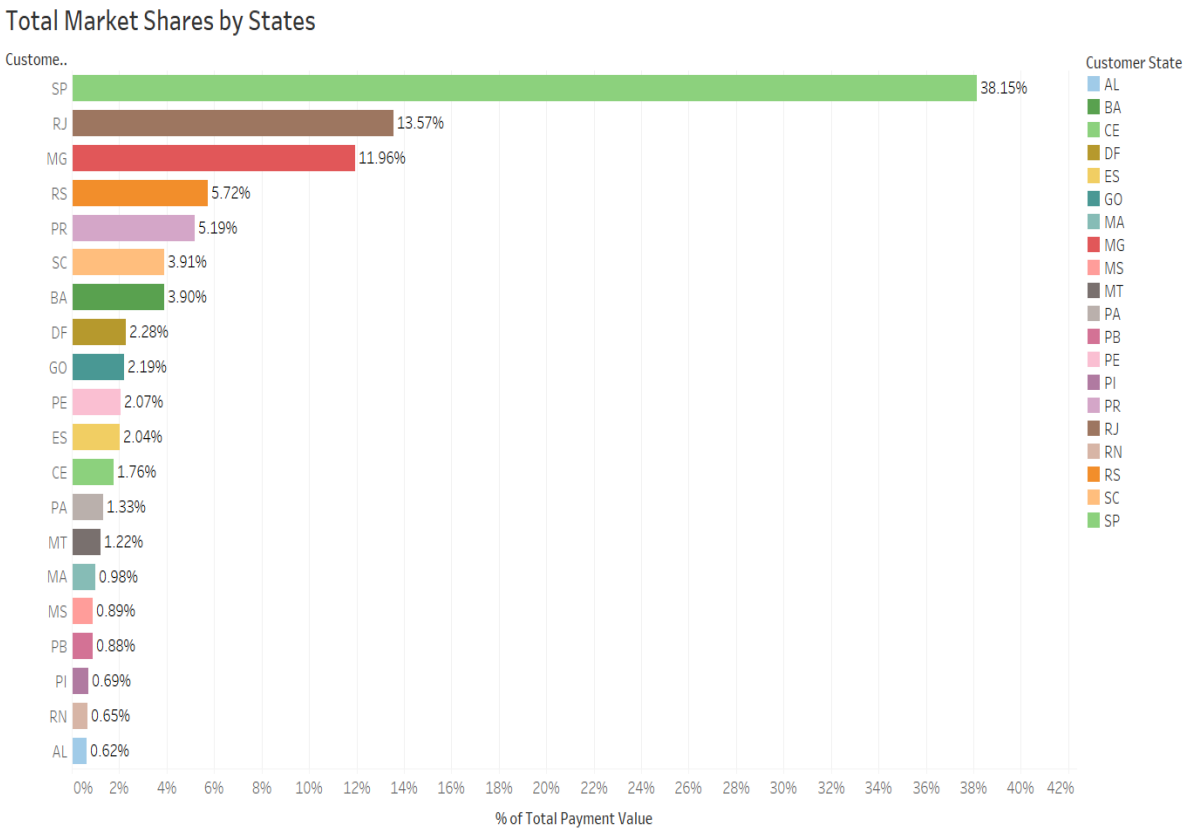
6.1 Best Business Day of Week



From the above graph it is very evident that weekdays seem to days where the business is best. Monday has the highest gross orders which drops marginally on Tuesdays and consecutively on Wednesday. The pattern showcases that maximum people prefer shopping on weekdays instead of weekend.

- **Order share on Weekdays: 77.02%**
- **Order share on weekends: 22.98%**

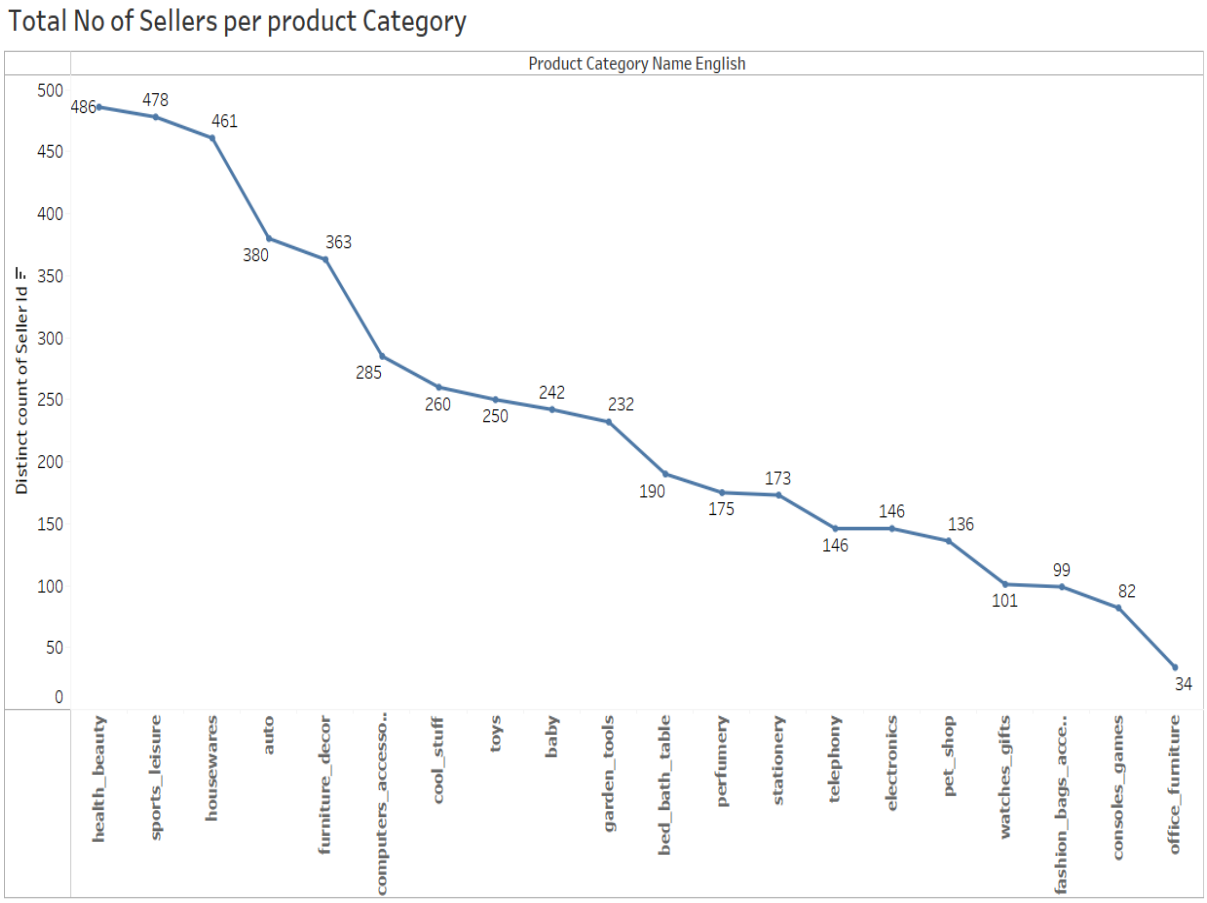
6.2 Total Revenue Share by States



% of Total Payment Value for each Customer State. Color shows details about Customer State. The data is filtered on Payment Type, which keeps boleto, credit_card, debit_card and voucher. The view is filtered on Customer State, which keeps 20 of 27 members.

Total revenue for Olist in the provided data from the duration Oct'2016 to Sep'2018 sums up to 14141991.32 Brazilian Real. Our of which **98.41% market share is captured by top twenty** states showcased in the graph above while **Sao Paula leads the way contributing around 38.5%** of overall Olist revenue. The state is largely populated and hence one of the prime reasons for contribution towards maximum orders and revenue.

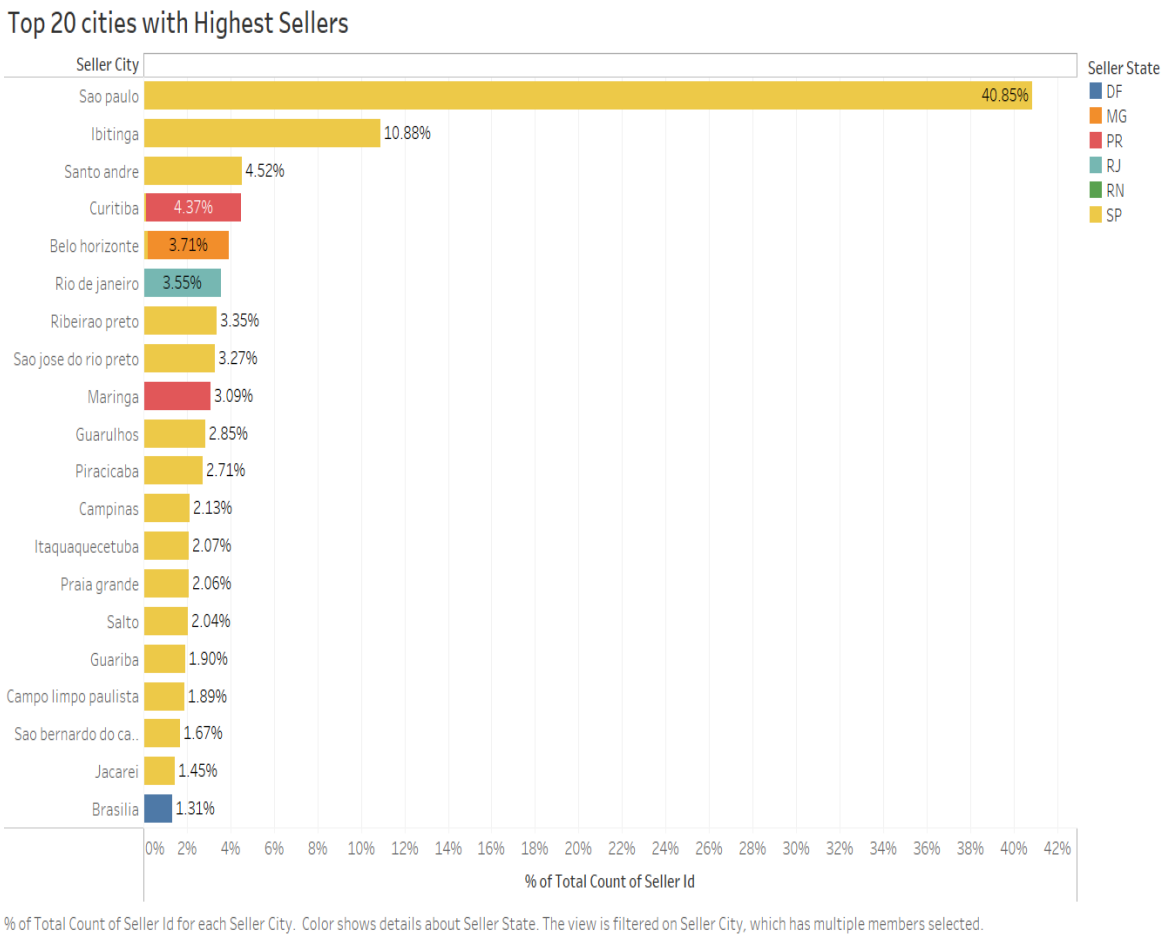
6.3 Total Number of Sellers per Category



The trend of distinct count of Seller Id for Product Category Name English. The data is filtered on sum of Payment Value, which keeps all values. The view is filtered on Product Category Name English, which keeps 20 of 71 members.

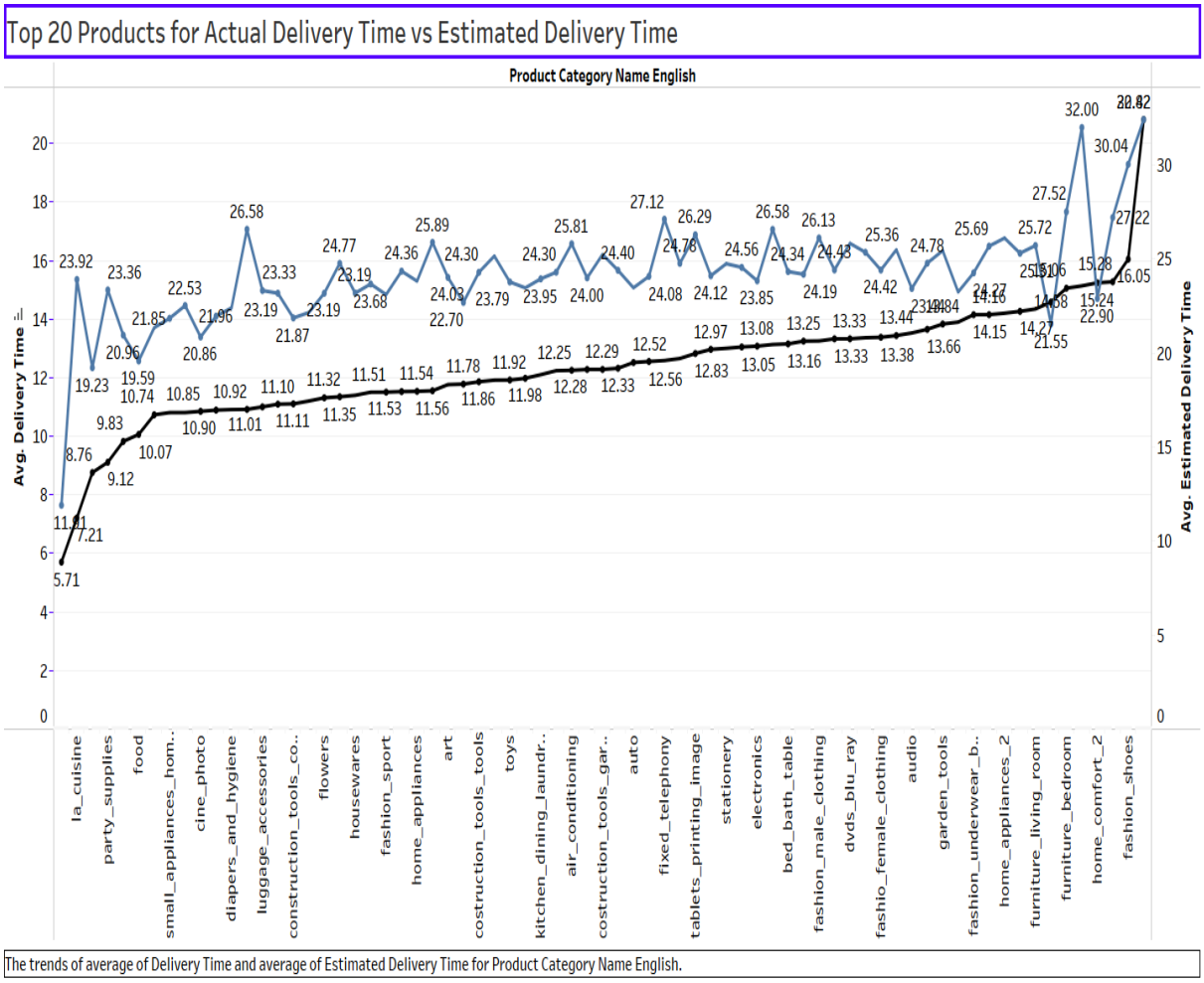
From the vast range of product offering, the most common product sellers listed on Olist for **Health, Sports, Beauty, House décor, Automobile Furniture**. From seller counts and Product category seller plot, we state that most seller prefers to sell product which have better user reviews and costing to **ensure bulk order and high profit margins**. At the same time, Office furniture sellers are very less leading to **high costing and high delivery time**, but these two parameters have marginal impact on the **user reviews** so we can assume that the **quality of the product delivered is compensating** for the above.

6.4 Top 20 cities with Highest Sellers



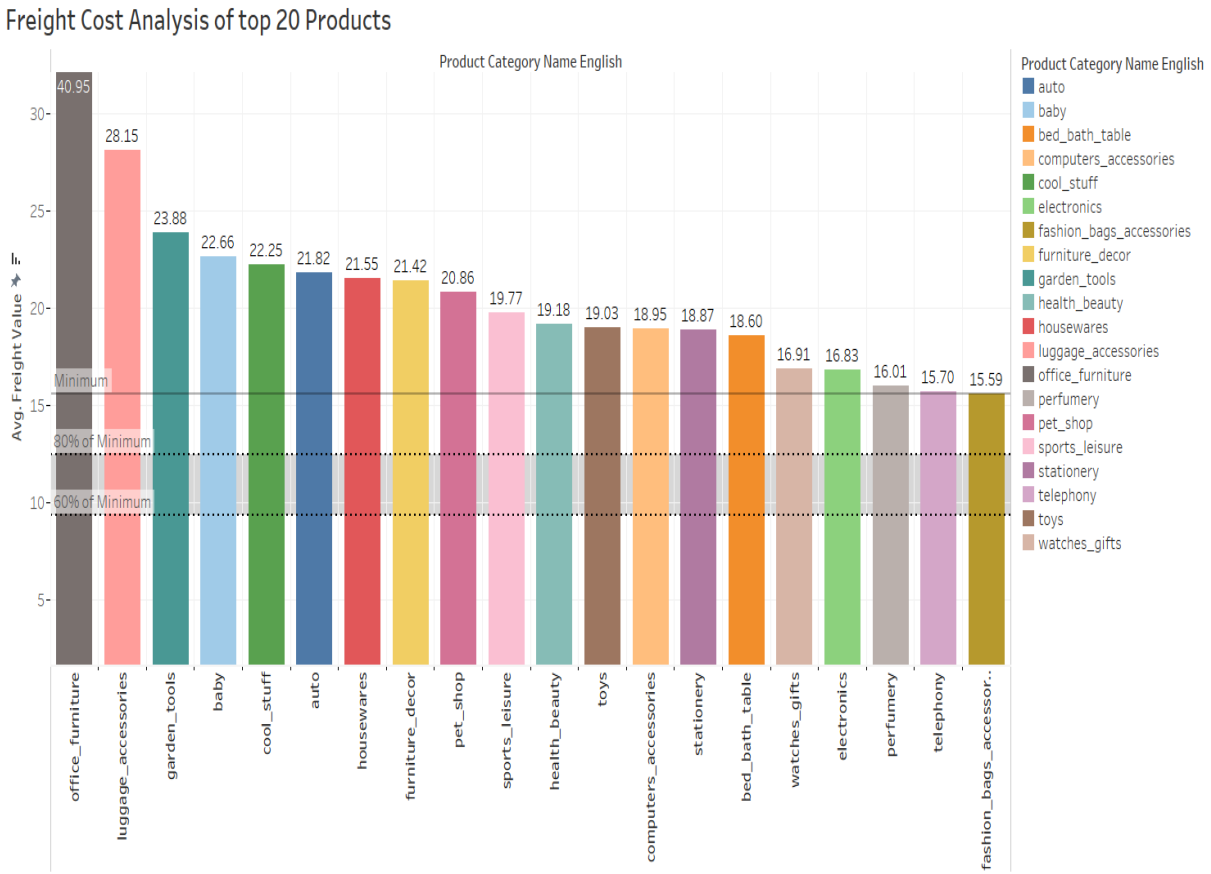
Sao Paula the state with **highest contribution** towards Olist **revenue**. So, it is evident that based on huge population and large orders, the state has maximum numbers of sellers which almost equal to the sellers with considering all other states. Sellers in Sao Paula are 90% whereas all other states comprise of around 10% only. Being a metro and from business perspective setting up your warehouse in Sao Paula can be a better idea.

6.5 Product Delivery Performance for top 20 products



For the Top 20 products, we have plotted the comparison of actual delivery time vs estimated delivery time. We observe that there is a clear deviation between the estimated and actual delivery time. We can say here that the higher the delivery time is the higher the customer will be dissatisfied with the service and higher possibility of a negative review. E.g., for diapers and hygiene, we see estimated delivery time was ~11 days while actual delivery is done in ~22 days. Delivery time is almost double here, for such a product customer would not want to wait this longer and would switch to other retailers which will lead to higher customer churning.

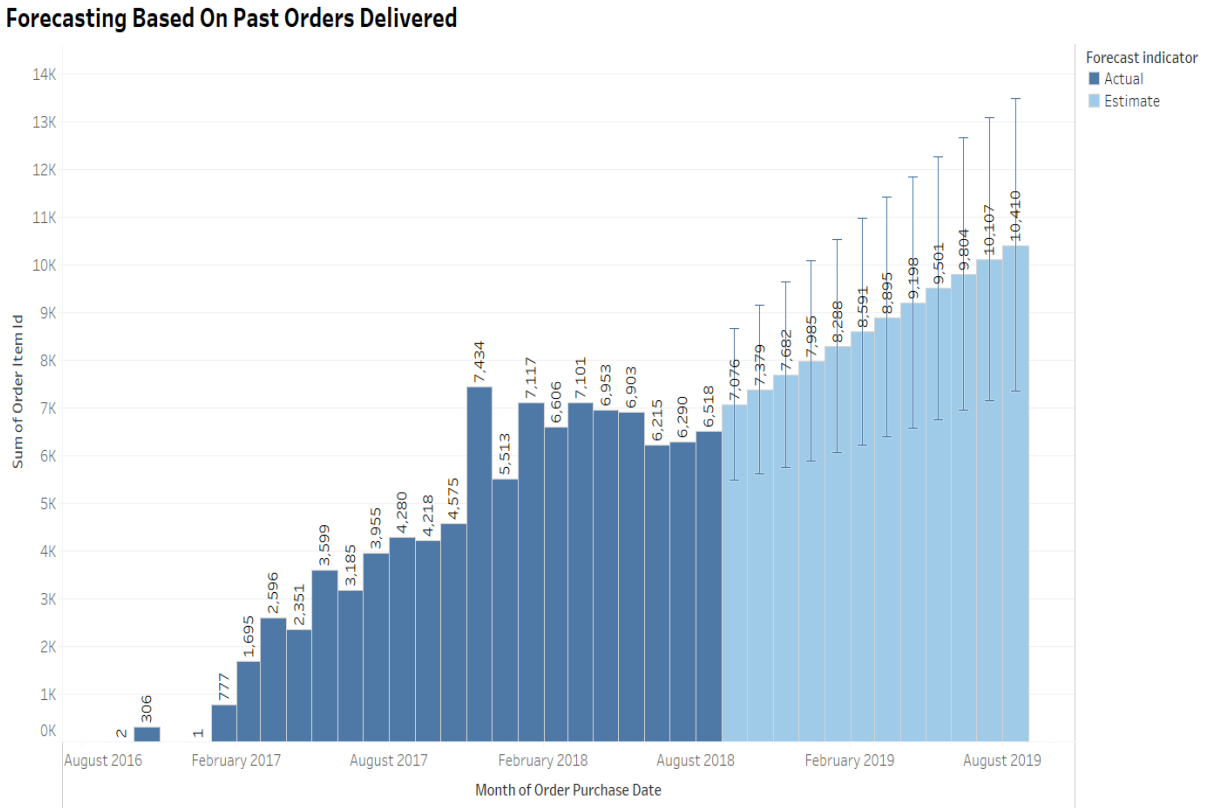
6.6 Freight Cost Analysis of top 20 products



Average of Freight Value for each Product Category Name English. Color shows details about Product Category Name English. The view is filtered on Product Category Name English and average of Freight Value. The Product Category Name English filter keeps 20 of 71 members. The average of Freight Value filter includes everything.

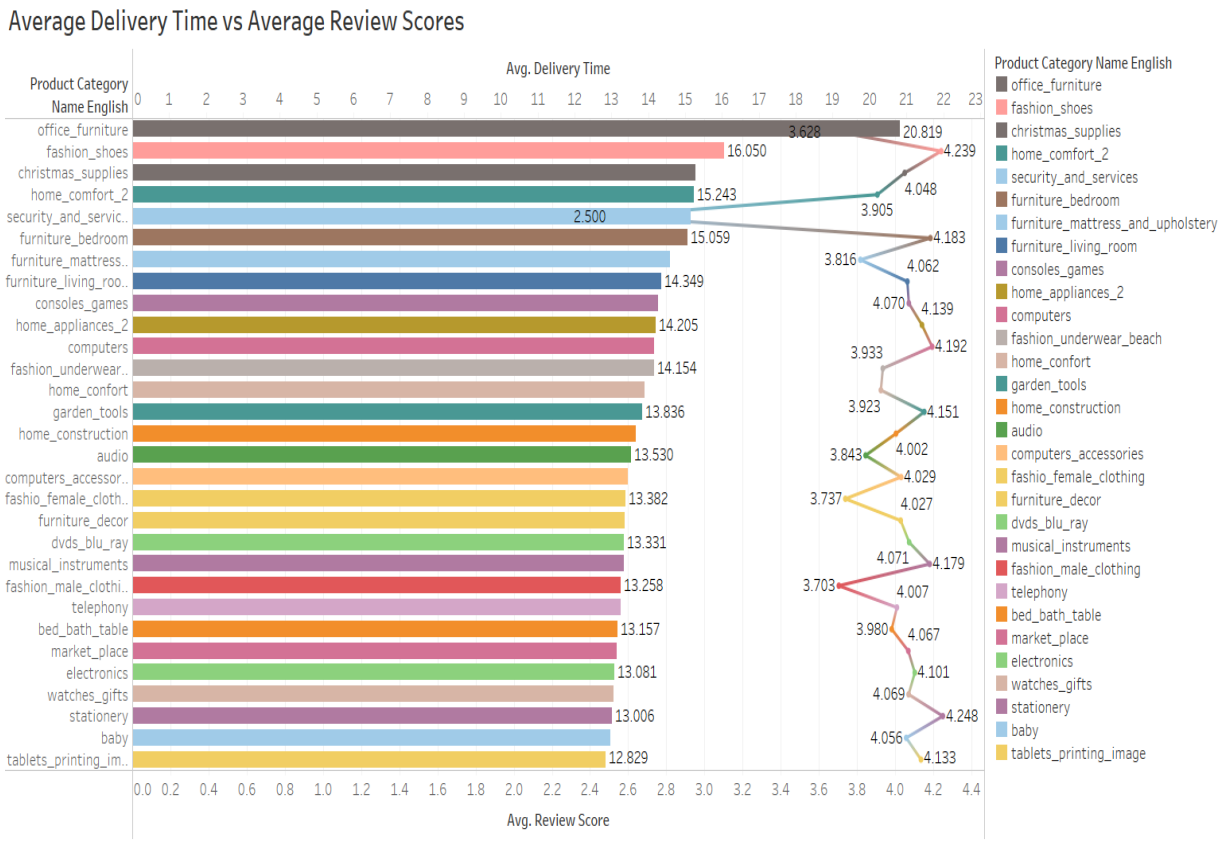
Here we plotted the average freight cost of the top 20 products. We see here that freight cost ranges from 15.59 for fashion bags accessories to 40.95 for office furniture. Usually, Office furniture has more than double the cost compared to fashion bags due to the size and weight of the office furniture. We can conclude that the bigger the size of the product the higher will be the freight cost for delivery.

6.7 Monthly Orders and sales forecasts



Based on the past orders history we have tried to predict the future orders. Even though the number of orders has flattened from February 2018 to August 2018, we expect the orders to pick up with some deviation from the predicted orders for the next 1 year. Olist here can improve the orders further by improving its delivery service with faster and accurate delivery of the products. As per current forecast, we expect August 2019 to have 60% more orders vs August 2018 with gradual rise every month.

6.8 Average delivery time vs Average review scores

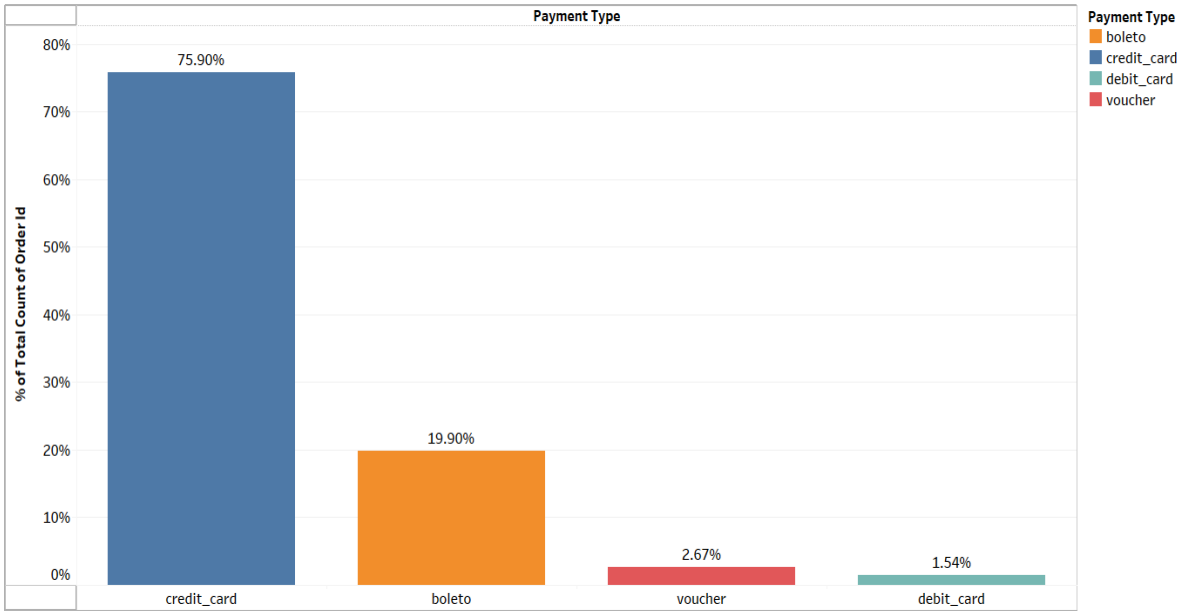


The trends of average of Review Score and average of Delivery Time for Product Category Name English. Color shows details about Product Category Name English. The view is filtered on Product Category Name English, which keeps 30 of 71 members.

We have plotted 30 product category's average delivery time & their customer review score on same scale to check if there is any correlation between delivery time and review score. It is also reported that 60% of product categories have average review scores above 4 out of 5 & their delivery time is average 13.5 days. Office Furniture, Fashion shoe and Christmas supply have reported 21 ,16 and 15 days respectively which were sold in year end. On other hand, table printing image, baby toys, stationery and electronics have reported 13 days delivery time with average review score of 4.1. We can also conclude that except furniture, security and service, product review scores and average delivery times nearly independent to each other.

6.9 Payment Methods used by Customer

Payment Method Used by Customers

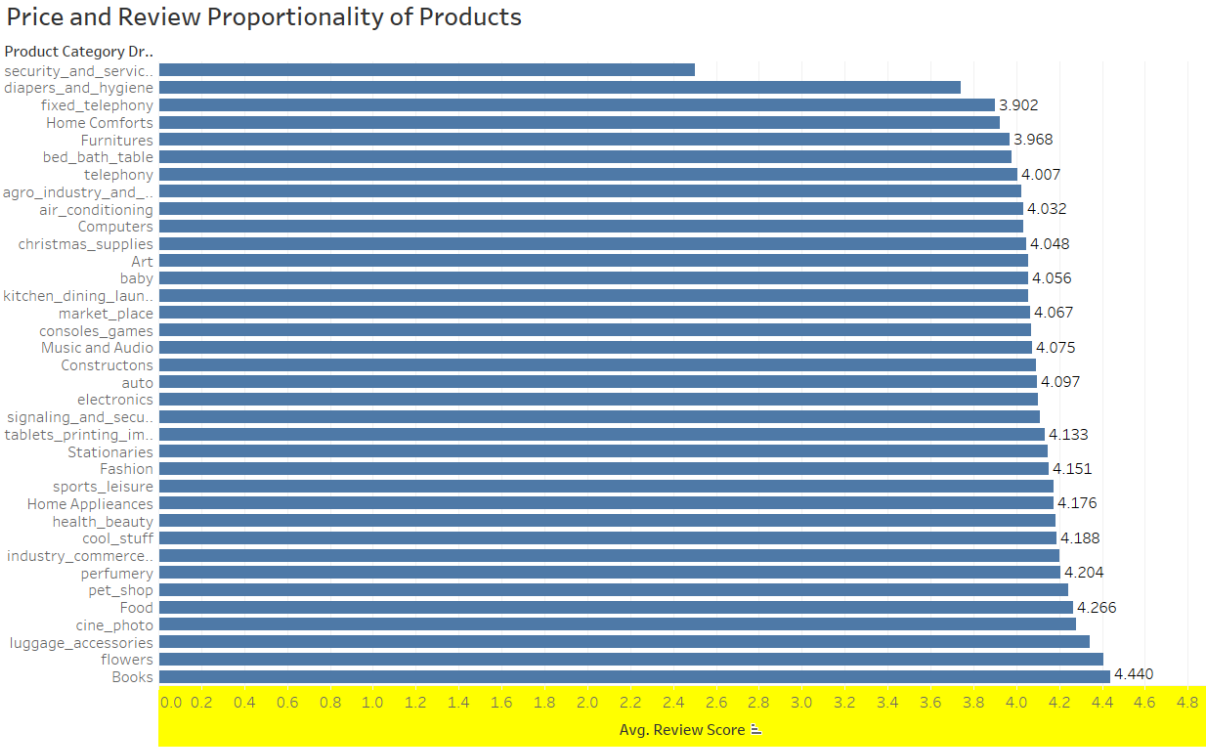


% of Total Count of Order Id for each Payment Type. Color shows details about Payment Type.

We have plotted the customer payment database and no of orders on the bar chart to check which payment mode is preferred by the customer. From the bar chart, it is reported **that a total of four types of payment methods** are used by customers i.e., credit card, Boleto, debit card and voucher. Out of all orders, **75.87% of orders** have reported **credit card** as payment mode **& 20% orders** reported **as Boleto payment** mode which contribute 95% of total payment records. We can conclude that Olist should use **credit card promotional schemes** to increase orders in upcoming years.

6.10 Price and review proportionality of products

This chart is based on Average Review Score for each product category. This view is filtered on average of review scores which range from 1.00 to 4.51. We can infer that the top most product categories are Books and flowers & people like it **4.44 out of 5**. & Lowest product category are Security & services product & people marked it **3.74 out of 5.** We can say that Books, Fashion & beauty products people like very much as compared to other products. Daily used product & luxury product category scores are higher as compared than other products. From marketing perspectives, we can target high review products & increase sales revenues.

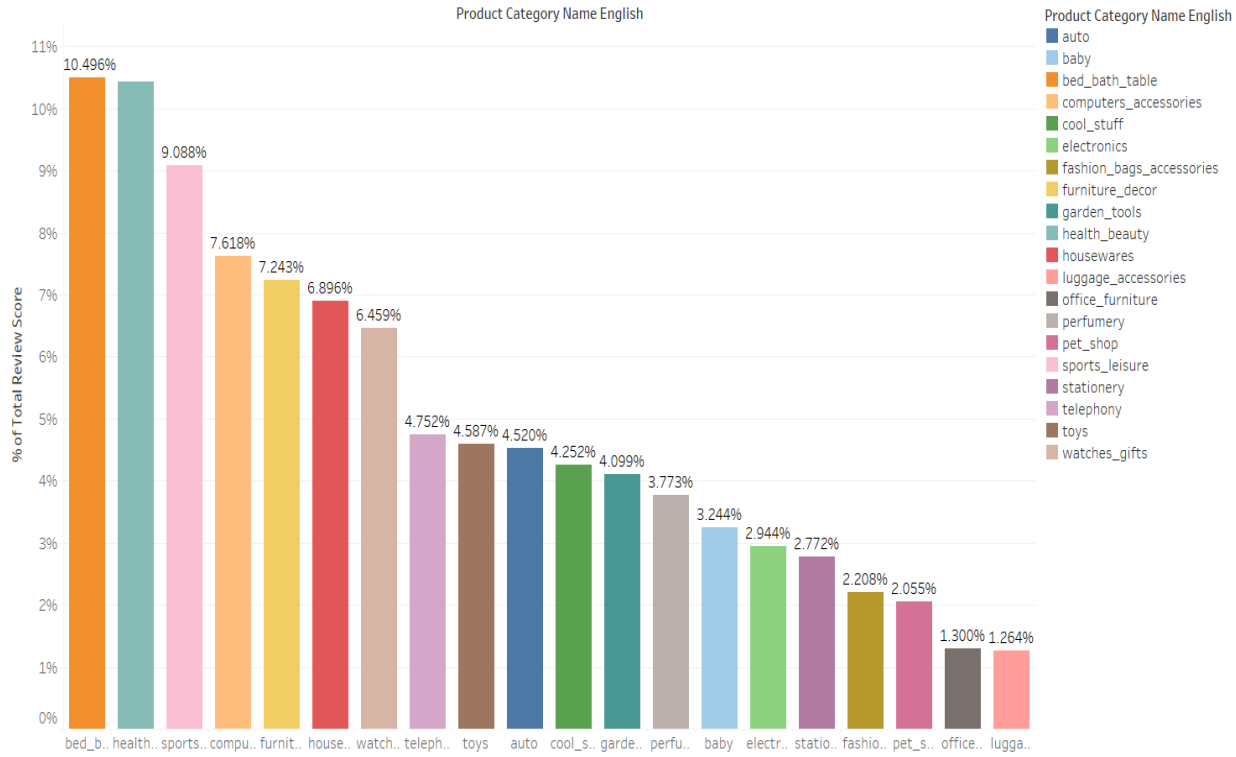


Average of Review Score for each Product Category Driven. The view is filtered on average of Review Score, which ranges from 1.000 to 4.510.

6.11 Review based Popular products

As we look at the Product Popularity Bar chart, it is reported that **BED BATH TABLE** is the top most popular product & has an average **10.50**% review score. Likewise, **LUGGAGE ACCESSORIES is the least** popular product with an average **1.26**% review score. Telephony, Toys & auto are in range of **4.52% to 4.72%. Luxury products are highly acceptable by people.** Top 10 popular product categories with average review scores reported between 4.5% to 10.5% can be targeted for product promotions to increase sales performance.

Product Popularity based on user reviews



% of Total Review Score for each Product Category Name English. Color shows details about Product Category Name English. The view is filtered on Product Category Name English, which keeps 20 of 71 members.

7. Conclusion Notes

- The variables provided and their respective class/labels are imbalanced. So it has been standardized before performing analysis.
- After performing the analysis, we can see that the most popular product categories based on user reviews are Bed Bath, Health, and Sports for the provided data set of duration Oct'16 to Sep'18.
- User reviews have been below average for security services whereas overall quality of all other products has been above good which is around user review 4.
- Preferred mode of transaction is credit card with around 76% users prefer this mode along with 20% people prefer another very similar mode called boleto i.e payment using vouchers and tickets accepted by central bank of Brazil. The Number of sellers has been very low for furniture products leading to high delivery time considering the demand.
- Sao Paula has been the major contributor considering revenue, User base, Sellers or even user reviews. Most of the Olist revenue is being generated from the location where most people prefer to shop on weekdays and enjoy their weekends.
- Driving new features was prime requirement specially to perform time series analysis, revenue comparison over month/weeks/days.

- Many numerical variables such as price, payment value, freight value have been useful for this analysis in comparing product accordingly and impact of same on user behaviors.
- Olist has been very focused on customer satisfaction considering the good user reviews across most of the product categories.
- The data needs to be cleaned and standardized first to ensure better analysis and in order to make sure that we are moving in a right direction we should identify the following:
    - What are we looking for?
    - Can I drive more features, or can I convert any variable from categorical or numerical?
    - What is the best tool which I can use to perform my analysis? Do I have knowledge and resources to use this tool?
    - What is the conclusion which will justify my analysis?

Video Presentation Link :-

Reference Links:

https://github.com/ricardozacarias/brazilian-ecommerce#Project-Abstract

https://github.com/jahoy/Brazilian-Ecommerce-data-analysis

https://github.com/VictorGuedes/Brazilian-E-Commerce-Public-Dataset-examples

https://medium.com/mlpoint/exploratory-data-analysis-eda-on-olist-dataset-10c8390b062f

https://jovian.ai/paritosh/edafinalb

https://github.com/ricardozacarias/brazilian-ecommerce#Project-Abstract

https://github.com/jahoy/Brazilian-Ecommerce-data-analysis

https://github.com/VictorGuedes/Brazilian-E-Commerce-Public-Dataset-examples

https://www.kaggle.com/thiagopanini/e-commerce-sentiment-analysis-eda-viz-nlp/notebook

https://www.analyticsvidhya.com/blog/2021/06/exploratory-data-analysis-using-data-visualization-techniques/

https://owenhsu94.medium.com/analysis-of-brazilian-e-commerce-datasets-olist-a33e38f677ea

https://medium.com/hamoye-blogs/unraveling-brazilian-e-commerce-dataset-e78463d77340

https://github.com/HamoyeHQ/g04-brazillian-commerce/tree/53108b4239f38de7650c220229c51ff57c7c17b2

https://github.com/yamenkaba/Brazilian-E-Commerce-Public-EDA-/blob/master/.ipynb_checkpoints/Brazilian%20E-Commerce%20Public(EDA)-checkpoint.ipynb

https://medium.com/analytics-vidhya/brazilian-e-commerce-public-eda-b8d02edd9aaf