Retail Store Analysis

Project Motivation

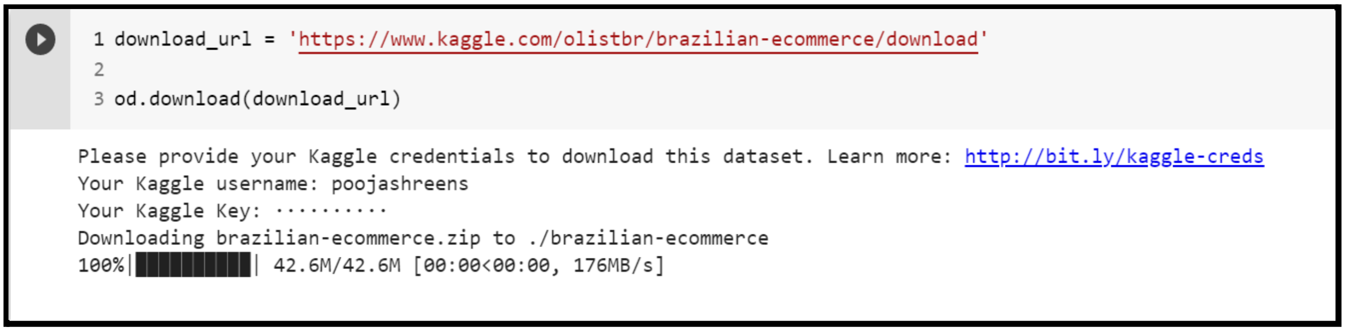
Clustering:
Customer segmentation and to get a better understanding of the group of customers who are at risk of churn so that retail stores can proactively engage with such customers to retain them.

Classification:
Upon analyzing and segmenting the customers based on their buying pattern, we are classifying customers based on their total worth to a business over the whole period of their relationship (Customer Lifetime Value)

Regression:
Predicting the product delivery date, a factor that influences the most to keep the customer satisfied.

Dataset

The dataset contains information of 100k orders placed in numerous Brazilian markets between 2016 and 2018. This is real business information that has been anonymized. Olist, the largest department store in Brazil, has generously shared this data collection. Olist connects small businesses from all around Brazil to the channel in a simple and cost-effective manner. The merchant can sell their products on the Olist Store and have them transported straight to customers through the Olist logistics partner. The vendor will be alerted to complete the order once the buyer has purchased the product from the Olist Store.When the client receives the product or the estimated delivery date approaches, the customer receives an email with a satisfaction survey where he or she can leave a note and some remarks on the purchase experience.
[Brazilian Olist Retail store dataset]



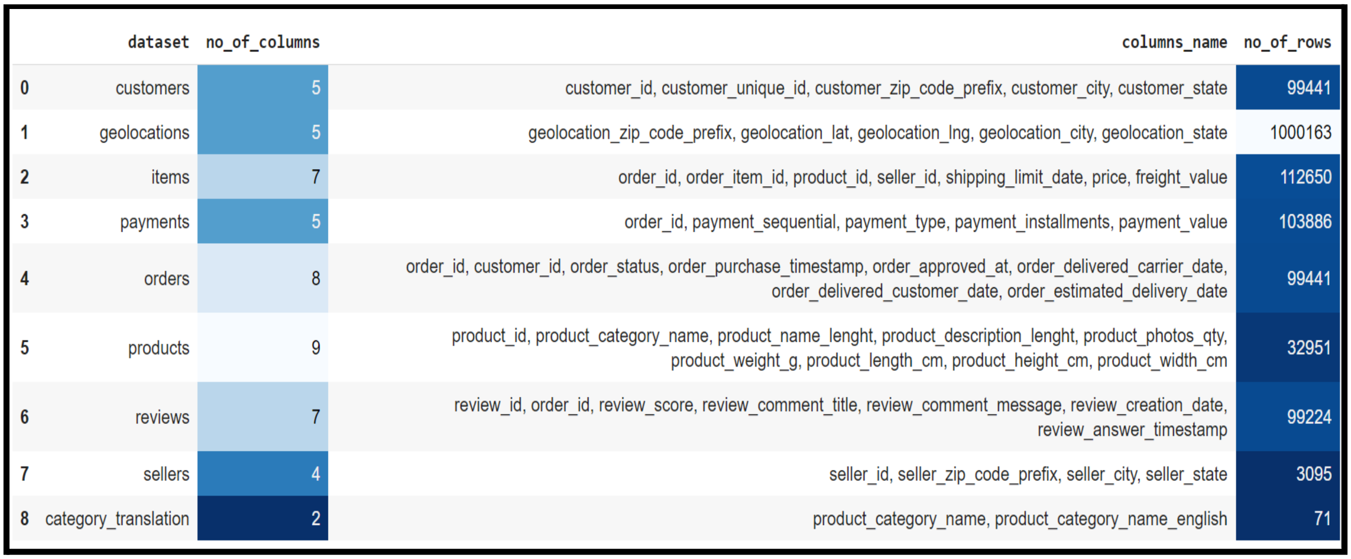
```
download_url = 'https://www.kaggle.com/olistbr/brazilian-ecommerce/download'

od.download(download_url)
```

```
Please provide your Kaggle credentials to download this dataset. Learn more: http://bit.ly/kaggle-creds
Your Kaggle username: poojashreens
Your Kaggle Key: ··········
Downloading brazilian-ecommerce.zip to ./brazilian-ecommerce
100%|██████████| 42.6M/42.6M [00:00<00:00, 176MB/s]
```

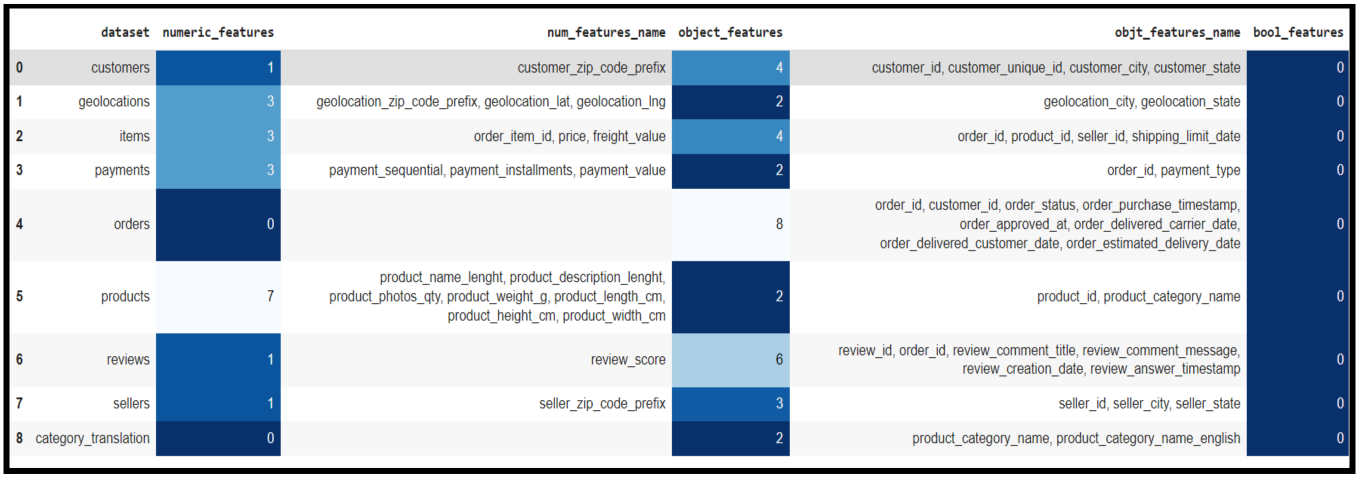
EDA

Checking number of rows and columns in all datasets:



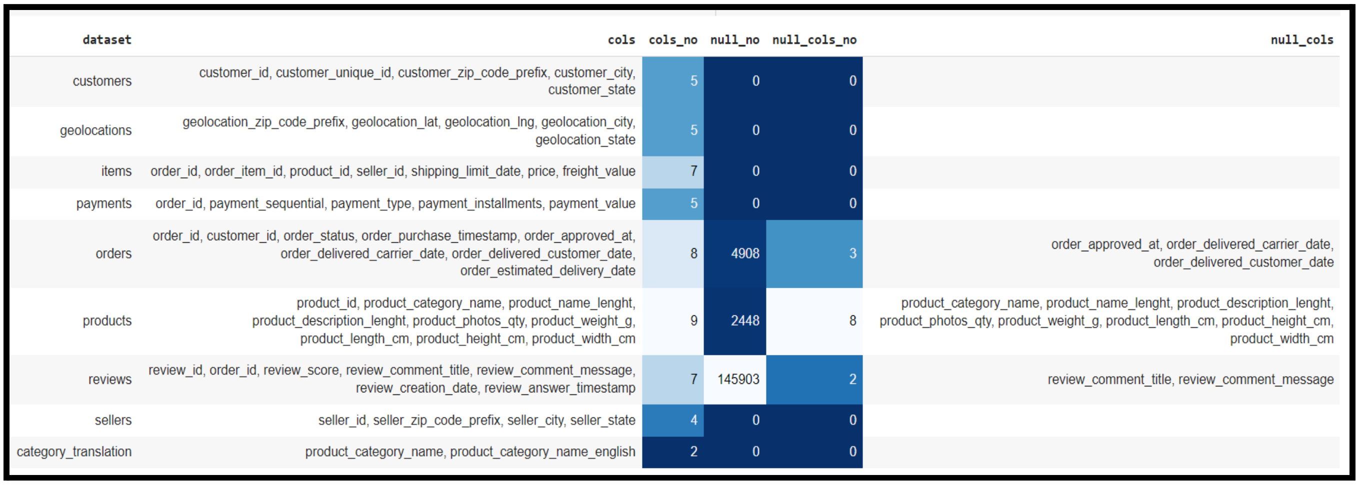
	dataset	no_of_columns	columns_name	no_of_rows
0	customers	5	customer_id, customer_unique_id, customer_zip_code_prefix, customer_city, customer_state	99441
1	geolocations	5	geolocation_zip_code_prefix, geolocation_lat, geolocation_lng, geolocation_city, geolocation_state	1000163
2	items	7	order_id, order_item_id, 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

Understanding dtypesin dataset:



	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	geolocation_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

Checking number of null values in dataset:



dataset	cols	cols_no	null_no	null_cols_no	null_cols
customers	customer_id, customer_unique_id, customer_zip_code_prefix, customer_city, customer_state	5	0	0	
geolocations	geolocation_zip_code_prefix, geolocation_lat, geolocation_lng, geolocation_city, geolocation_state	5	0	0	
items	order_id, order_item_id, product_id, seller_id, shipping_limit_date, price, freight_value	7	0	0	
payments	order_id, payment_sequential, payment_type, payment_installments, payment_value	5	0	0	
orders	order_id, customer_id, order_status, order_purchase_timestamp, order_approved_at, order_delivered_carrier_date, order_delivered_customer_date, order_estimated_delivery_date	8	4908	3	order_approved_at, order_delivered_carrier_date, order_delivered_customer_date
products	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	9	2448	8	product_category_name, product_name_lenght, product_description_lenght, product_photos_qty, product_weight_g, product_length_cm, product_height_cm, product_width_cm
reviews	review_id, order_id, review_score, review_comment_title, review_comment_message, review_creation_date, review_answer_timestamp	7	145903	2	review_comment_title, review_comment_message
sellers	seller_id, seller_zip_code_prefix, seller_city, seller_state	4	0	0	
category_translation	product_category_name, product_category_name_english	2	0	0	

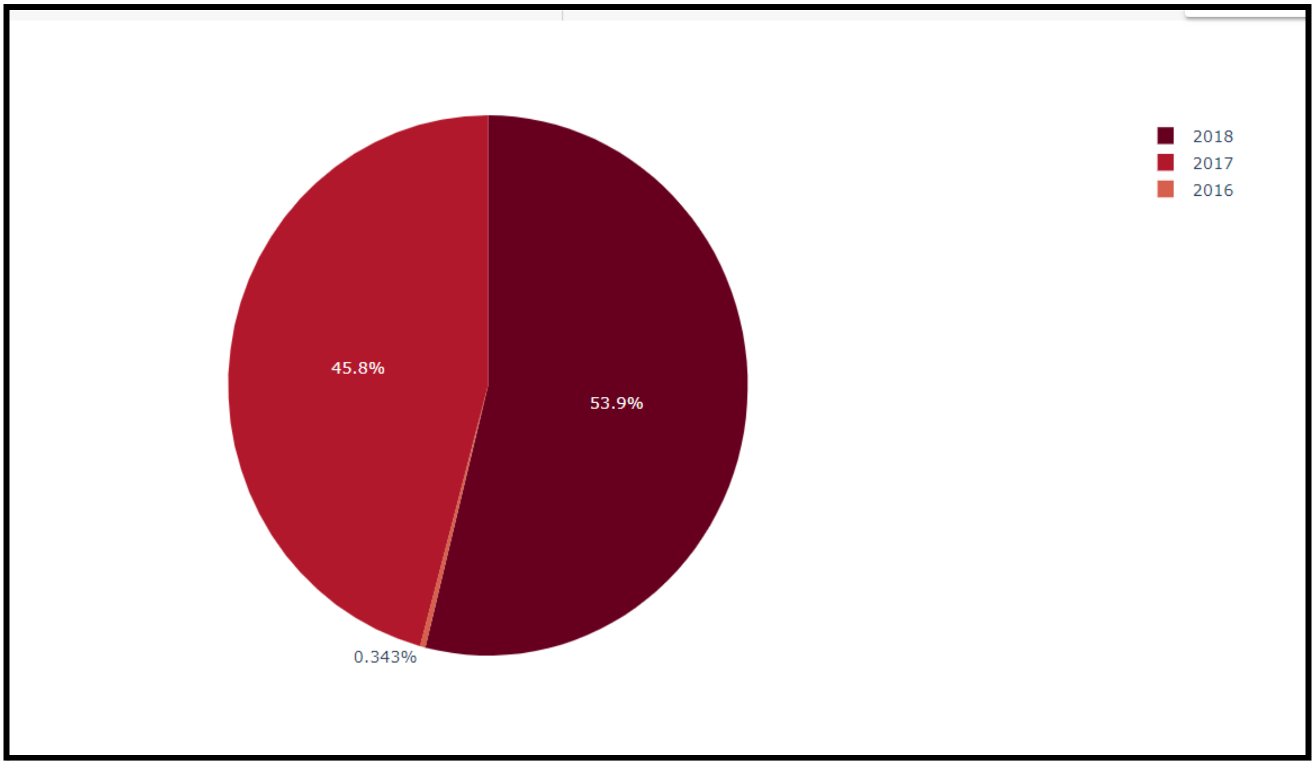
Handling Missing values in the dataset:

1. Timestamps containing missing values are order_approved_at, order_delivered_carrier_date, order_delivered_customer_date.
2. Null-values in order_approved_at can be replaced by order_purchase_timestamp and null-values in order_delivered_customer_date can be replaced by order_estimated_delivery_date
3. we can drop the column order_delivered_carrier_date.
4. Product related details can be filled by taking the median of those columns.
5. Review comments can be filled using the "No review" string.

Data Deduplication:

Eliminating duplicate entries in the data.

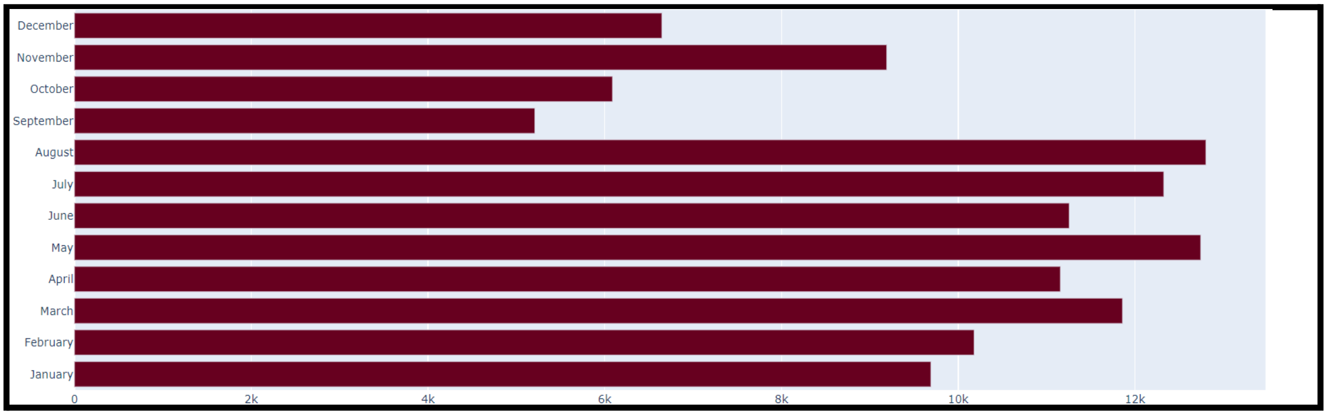
Converting timestamp into date and time format for data analysis:



Pie chart shows that there is increase in sales yearly from 2016 to 2018

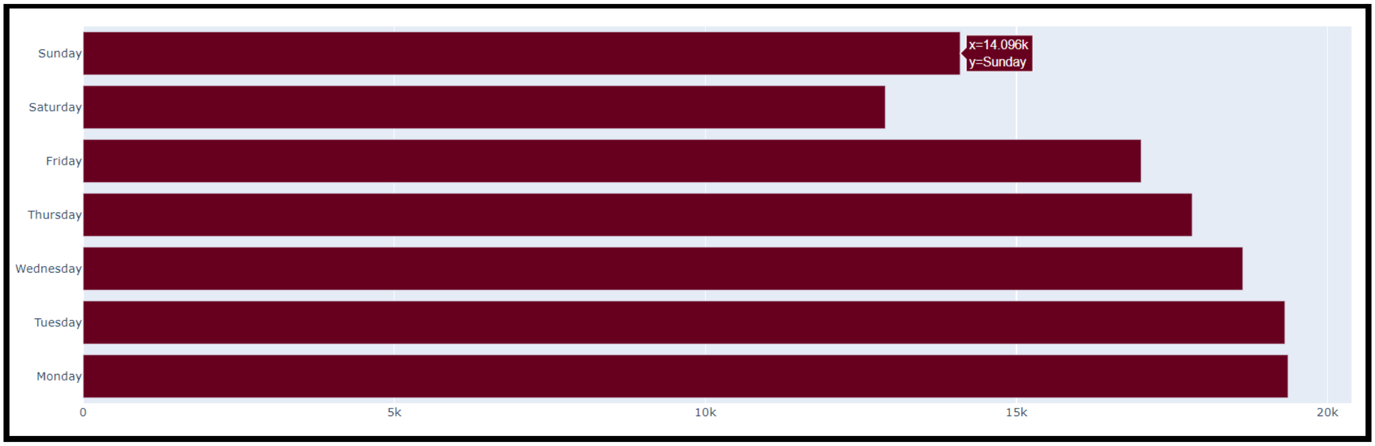
Data Analysis:

Analyzing sales monthly:



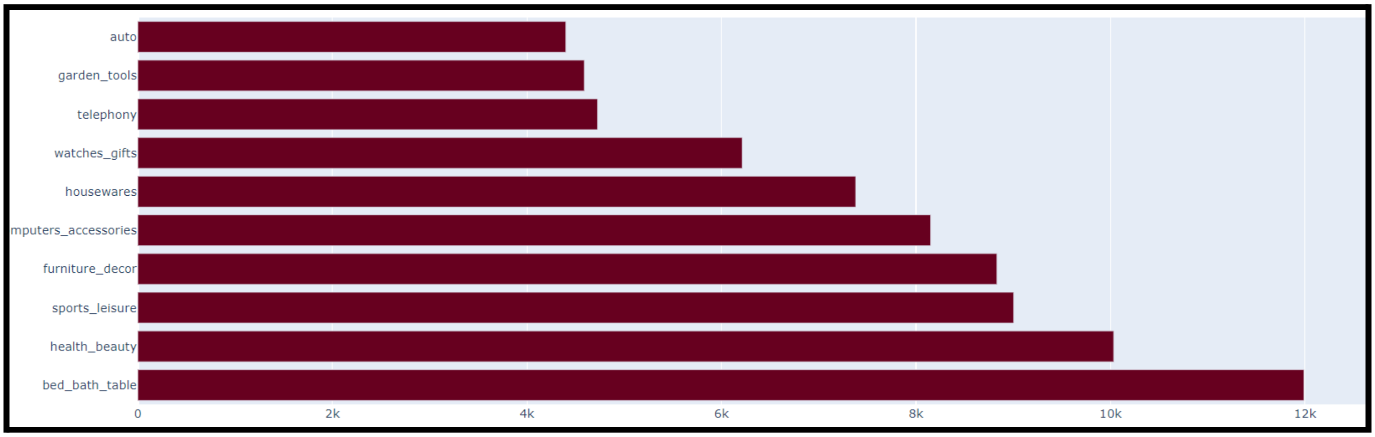
August month has the highest sales in Brazil

Analyzing sales weekly:



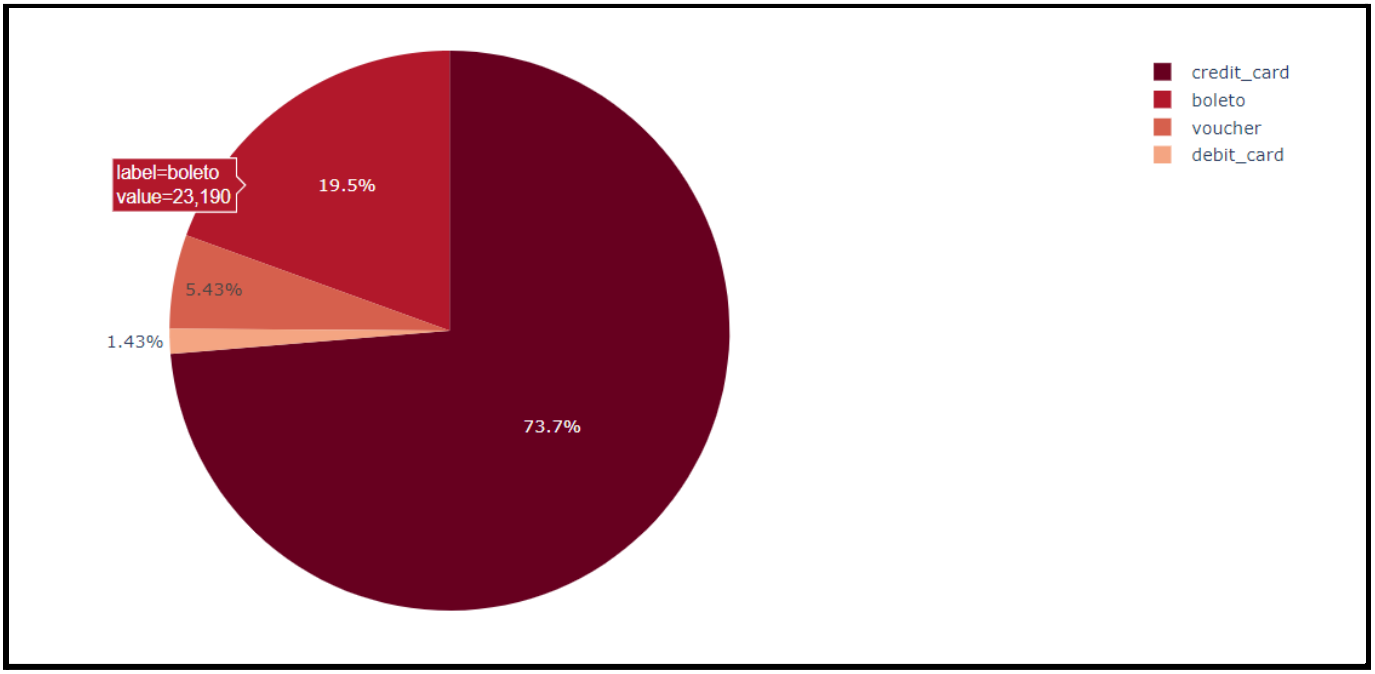
Monday is where most of the orders are placed by the customers.

Analyzing products sales:



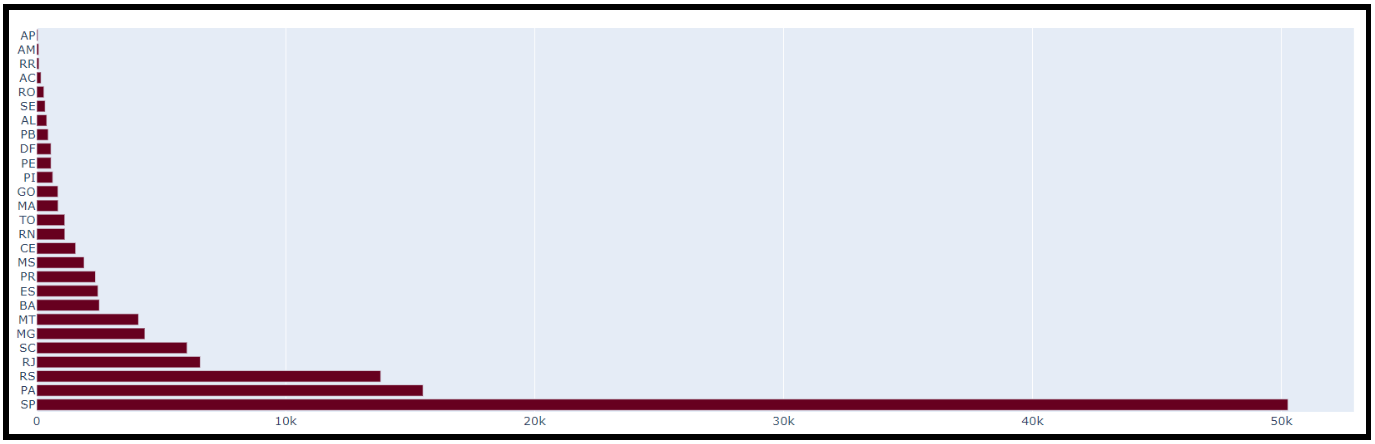
Bed_bath_table followed by health_beauty has the highest sales across Brazil.

Analyzing Payment Method:



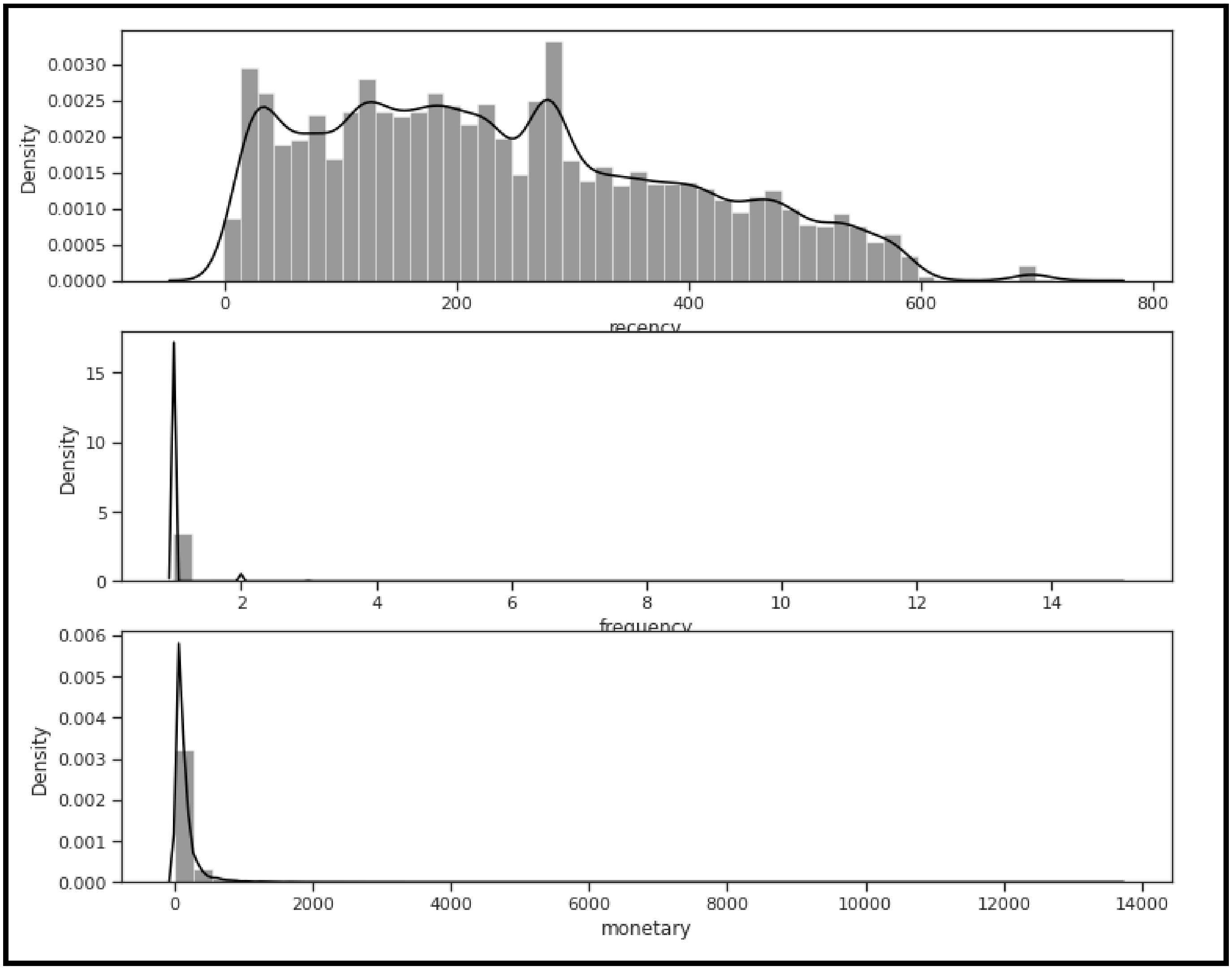
Almost 3/4th of the customers use credit cards to buy the products, offering discounts on credit cards will increase sales.

Analyzing customer count per state:



Feature Extraction

RFM analysis is a customer behavior segmentation technique. Based on customers' historical transactions, RFM analysis focuses on 3 main aspects of customers' transactions: **recency**, **frequency** and **purchase amount**. Understanding these behaviors will allow businesses to cluster different customers into groups.
Recency (R): How recently customers have made their purchases.
Frequency (F): How often customers have made their purchases.
Monetary (M): How much money customers have paid for their purchases.



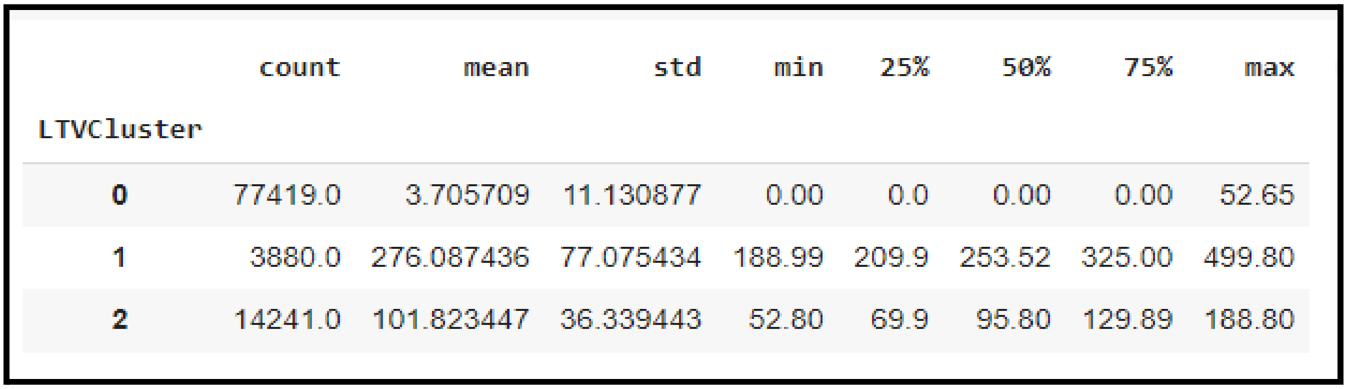
There are three density plots of recency, frequency and monetary are plotted.From the first plot of recency we can observe that most of the users stayed with olist for a long duration which is positive thing but order frequency is less.

From the second plot of frequency most number of transactions or orders is less than 5. from the third plot of monetary the maximum amount spent over the given very period is seems to be less than 1500 approx.

Segmenting customers using RFM analysis is the outcome of clustering. Using those features we are extracting a target feature called Customer Lifetime value.

LTVCluster Feature:

This feature is extracted by clustering customers based on their revenue from the past 6 months and that is used as a target feature.
LTVCluster tells us if the customer is valuable for our business or not.



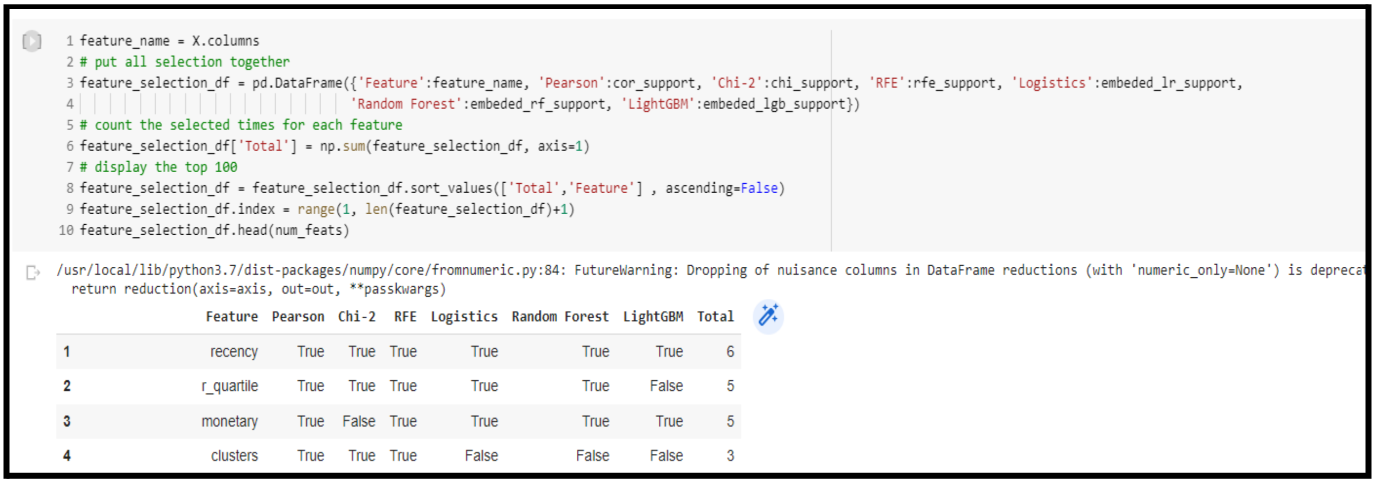
	count	mean	std	min	25%	50%	75%	max
LTVCluster								
0	77419.0	3.705709	11.130877	0.00	0.0	0.00	0.00	52.65
1	3880.0	276.087436	77.075434	188.99	209.9	253.52	325.00	499.80
2	14241.0	101.823447	36.339443	52.80	69.9	95.80	129.89	188.80

Feature Selection

Important features are using the following Feature selection technique

- Pearson Correlation
- Chi-Squared
- Recursive Feature Elimination
- Lasso: SelectFromModel
- Tree-based: SelectFromModel

Using all technique together:



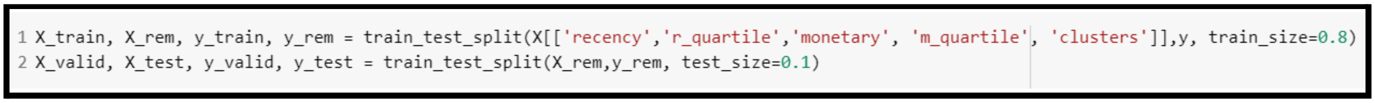
```
feature_name = X.columns
# put all selection together
feature_selection_df = pd.DataFrame({'Feature':feature_name, 'Pearson':cor_support, 'Chi-2':chi_support, 'RFE':rfe_support, 'Logistics':embeded_lr_support,
                                    'Random Forest':embeded_rf_support, 'LightGBM':embeded_lgb_support})
# count the selected times for each feature
feature_selection_df['Total'] = np.sum(feature_selection_df, axis=1)
# display the top 100
feature_selection_df = feature_selection_df.sort_values(['Total','Feature'] , ascending=False)
feature_selection_df.index = range(1, len(feature_selection_df)+1)
feature_selection_df.head(num_feats)
```

```
/usr/local/lib/python3.7/dist-packages/numpy/core/fromnumeric.py:84: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecat
  return reduction(axis=axis, out=out, **passkwargs)
```

	Feature	Pearson	Chi-2	RFE	Logistics	Random Forest	LightGBM	Total
1	recency	True	True	True	True	True	True	6
2	r_quartile	True	True	True	True	True	False	5
3	monetary	True	False	True	True	True	True	5
4	clusters	True	True	True	False	False	False	3

Splitting data:

In order to eliminate overfitting and underfitting the whole dataset is divided into train, test and validate.

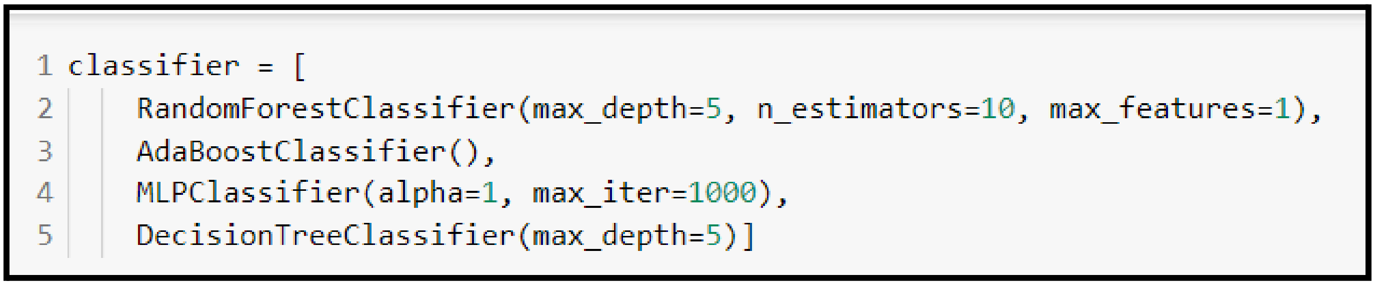


```
X_train, X_rem, y_train, y_rem = train_test_split(X[['recency','r_quartile','monetary', 'm_quartile', 'clusters']],y, train_size=0.8)
X_valid, X_test, y_valid, y_test = train_test_split(X_rem,y_rem, test_size=0.1)
```

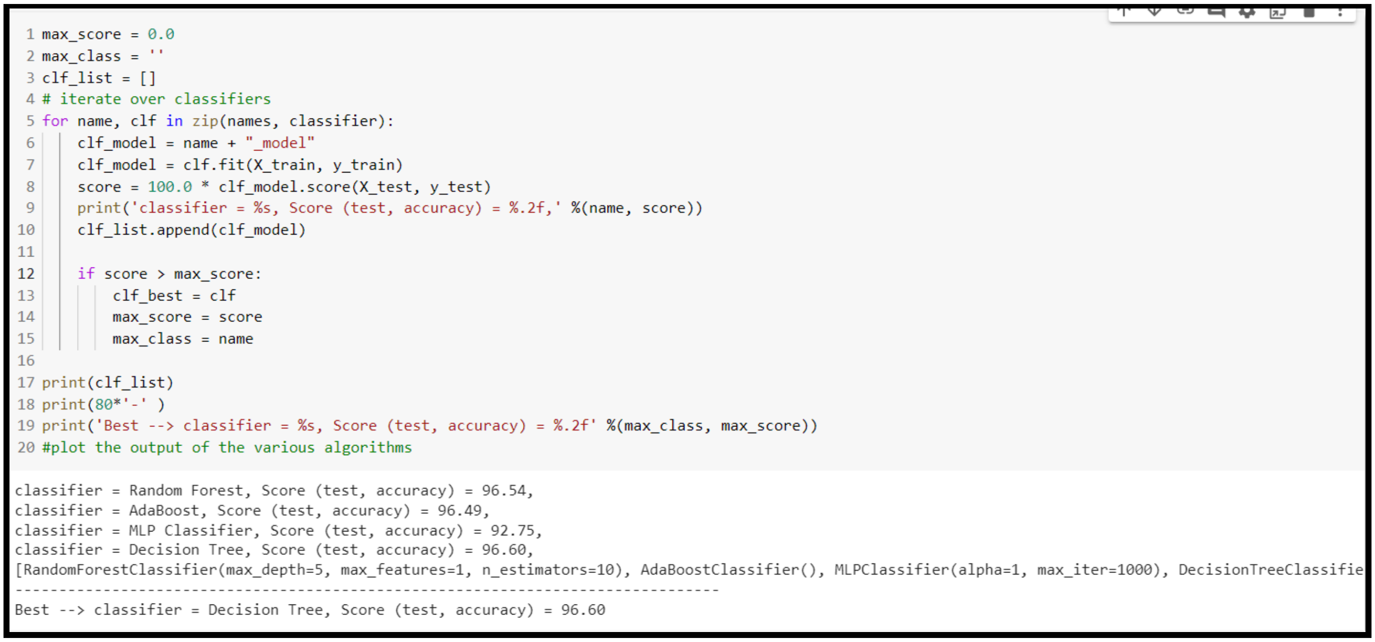
Modeling:

classification refers to a predictive modeling problem where a class label is predicted for a given example of input data.

Implementation of classier model using muller loop.



```
classifier = [
    RandomForestClassifier(max_depth=5, n_estimators=10, max_features=1),
    AdaBoostClassifier(),
    MLPClassifier(alpha=1, max_iter=1000),
    DecisionTreeClassifier(max_depth=5)]
```



```
max_score = 0.0
max_class = ''
clf_list = []
# iterate over classifiers
for name, clf in zip(names, classifier):
    clf_model = name + "_model"
    clf_model = clf.fit(X_train, y_train)
    score = 100.0 * clf_model.score(X_test, y_test)
    print('classifier = %s, Score (test, accuracy) = %.2f,' %(name, score))
    clf_list.append(clf_model)

    if score > max_score:
        clf_best = clf
        max_score = score
        max_class = name

print(clf_list)
print(80*'-' )
print('Best --> classifier = %s, Score (test, accuracy) = %.2f' %(max_class, max_score))
#plot the output of the various algorithms
```

```
classifier = Random Forest, Score (test, accuracy) = 96.54,
classifier = AdaBoost, Score (test, accuracy) = 96.49,
classifier = MLP Classifier, Score (test, accuracy) = 92.75,
classifier = Decision Tree, Score (test, accuracy) = 96.60,
[RandomForestClassifier(max_depth=5, max_features=1, n_estimators=10), AdaBoostClassifier(), MLPClassifier(alpha=1, max_iter=1000), DecisionTreeClassifie
--------------------------------------------------------------------------------
Best --> classifier = Decision Tree, Score (test, accuracy) = 96.60
```

Performance Analysis:

F1 Score : The F1 score is a number between 0 and 1 and is the harmonic mean of precision and recall. It is a sort of maintains a balance between the precision and recall for your classifier