GREAT LAKES
INSTITUTE OF MANAGEMENT, CHENNAI

greatlearning
Learning for Life

Project Summary

Batch details	DSE-FT-CHN-JULY'22
Team members	Ahamed basha Anish Kumar K Arif Hussain K M. Khushi Rashika S
Domain of Project	Retail Analytics
Proposed Project Title	Product sales prediction and segmentation
Group Number	4
Team Leader	Anish Kumar K
Mentor Name	Vidhya Kannaiah

Date: 22-12-2022

Table of Contents

S. No	Topic	Page No
1	Introduction	3
2	Dataset Understanding	4
3	Data Pre-processing	6
4	Exploratory Data Analysis	8
5	Statistical Test	28
6	Segmentation	32
7	Model Building	33
8	Business Inference	36

INTRODUCTION:

Amazon is an e-commerce platform that sells many product lines, including media (books, movies, music, and software), apparel, baby products, consumer electronics, beauty products, gourmet food, groceries, health and personal care products, industrial & scientific supplies, kitchen items, jewelry, watches, lawn and garden items, musical instruments, sporting goods, tools, automotive items, toys and games, and farm supplies and consulting services.

In this dataset, we have amazon sales of 9 Brands, say,

- **BabyPro** - BabyPro is a brand that deals with baby-proofing products like corner guards, edge guards, socket covers, child locks and others.
- **Cinagro** - Cinagro is a brand which sells Gardening essentials
- **Frenchware** - Frenchware is an Online Products for Home and Kitchen.
- **Rolid** - Rolid is the brand which sells Electronic Home Safe / Locker for Home, Office, and Shops.
- **Rusabl** - Rusabl is the brand which sells Sustainable and Eco-Friendly Products.
- **Senego** - Senego is the brand that sells beanbags and Laptop sleeves etc.
- **Vifitkit** - Vifitkit is yoga mats manufacturer and supplier.
- **XTrim** - XTrim customizes development of Laces, Belts, Buttons, Metal Fittings & host of new fashion Trims.
- **Yogaraise** - Yogaraise is company which deals with the Yoga mats and products.

BUSINESS PROBLEM STATEMENT:

Business Problem Understanding:

The sales of each brand in amazon differs by order state, date, and other features. There are states where some brands have no sales and some brands find difficulties while selling their products and there are states where particular products are highly purchased too.

Business Objective:

The objective is to device a machine learning model to predict the total payment for each brand. The objective is to device a model to group the products based on their brands and the locality where they have been sold and prioritize the products based on the sales. Our objective is to find out which brands products is getting high sales.

Approach:

Our approach to the business problem involves Data understanding, Data Pre-processing, and Exploratory data Analysis (EDA) on the data that we have obtained, and understanding the dependency of features in the prediction of price and segmentation basis on brands and locality.

DATASET INFORMATION:

Dataset consists of the several variables providing information about the sales with 31,936 rows and 23 columns in the data. This is also the most vital set of information which would be used to segregate for train test split function later in order to check the accuracy.

DATA UNDERSTANDING:

- Dataset consists of the several variables providing information about the sales.
- It has 23 features and 31,936 records.

INDEPENDENT VARIABLES:

Variable Name	Variable Descriptions
date/time	Date and time in UTC
Settlement id	Id of the settlement
Type	Type of Order
Order id	Indicates unique identity number for the orders
Brand Name	Name of the Brand

Sku	Stock Keeping Units
Description	Overview of the Product
Quantity	Number of ordered quantity
Account Type	Mode of Transaction
Fulfillment	Amazon or Merchant
Order state	State of Order placed
Order postal	Postal code
Amazon sales	Sales in Amazon
Shipping Credits	Shipping Charges
Promotional Rebates	Promotional or discounted Price
Total sales tax liable (GST before adjusting TCS)	Total GST before TCS
TCS-CGST	Central Goods and Service Tax
TCS-SGST	State Goods and Service Tax
TCS-IGST	Integrated Goods and Service Tax
Selling Fees	Selling charges to the company
Fba Fees	Fulfillment Charges
Other Transaction Fees	Transaction Charges
Total Payment	Total amount of Amazon sales with the Fees and Tax

DEPENDENT VARIABLE (TARGET VARIABLE):

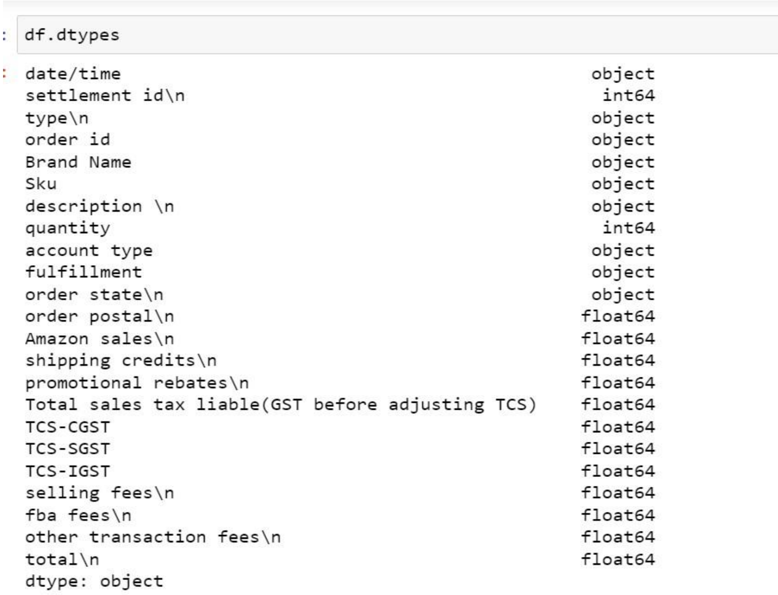
Total Payment – Total Payment is the amount that is made by the customers after tallying with the GST, Shipping Credits, Selling Fees, and Other Transaction Fees.

DATA PRE-PROCESSING:

- We checked for the dtypes of the dataset and found that the date time column is in object so we changed that column dtype as datetime
- In our dataset, we had around 111 duplicated records and we removed it for further analysis.
- Segregated Date and Time from Date Time variable and created new column as Date and Time separately.
- The Columns Order State and Order Postal has 998 null values and Order id has 22 null values and Fulfillment had 978 null values.
- We separated the data wherever we had null values in Order state and Order Postal and did analysis on it.
- Extracted the number of pieces from the Sku code and created a new column as ‘No. of Pieces’
- We merged Shipping Credits, Promotional Rebates, Total Sales Tax Liable (GST before adj TCS), TCS_CGST, TCS_SGST, TCS_IGST, Selling Fees, Fba Fees, Other Transaction Fees these variables as charges of the order into one variable to avoid multicollinearity.
- Created a new variable based on the Sku and transformed with five point summary statistics and saved it in new columns.
- Now the shape of data is (30827 Rows and 34 Columns)

DATA TYPE VERIFICATION:

The dataset which we opted is of more than 30 thousand rows and 23 columns in which we see missing values and treated it accordingly. Each columns data types are checked for the dataset and the data types are preprocessed and checked whether there is a mismatch of the datatypes.



```
: df.dtypes

: date/time                                           object
  settlement id\n                                      int64
  type\n                                              object
  order id                                            object
  Brand Name                                          object
  Sku                                                 object
  description \n                                      object
  quantity                                             int64
  account type                                        object
  fulfillment                                         object
  order state\n                                       object
  order postal\n                                     float64
  Amazon sales\n                                     float64
  shipping credits\n                                 float64
  promotional rebates\n                              float64
  Total sales tax liable(GST before adjusting TCS)   float64
  TCS-CGST                                           float64
  TCS-SGST                                           float64
  TCS-IGST                                           float64
  selling fees\n                                     float64
  fba fees\n                                         float64
  other transaction fees\n                           float64
  total\n                                            float64
  dtype: object
```

VARIABLE CATEGORIZATION:

Total Columns – 23

Total Rows – 31936

- **Independent Variables:**

    Numerical Columns – 14

    Categorical Columns – 8

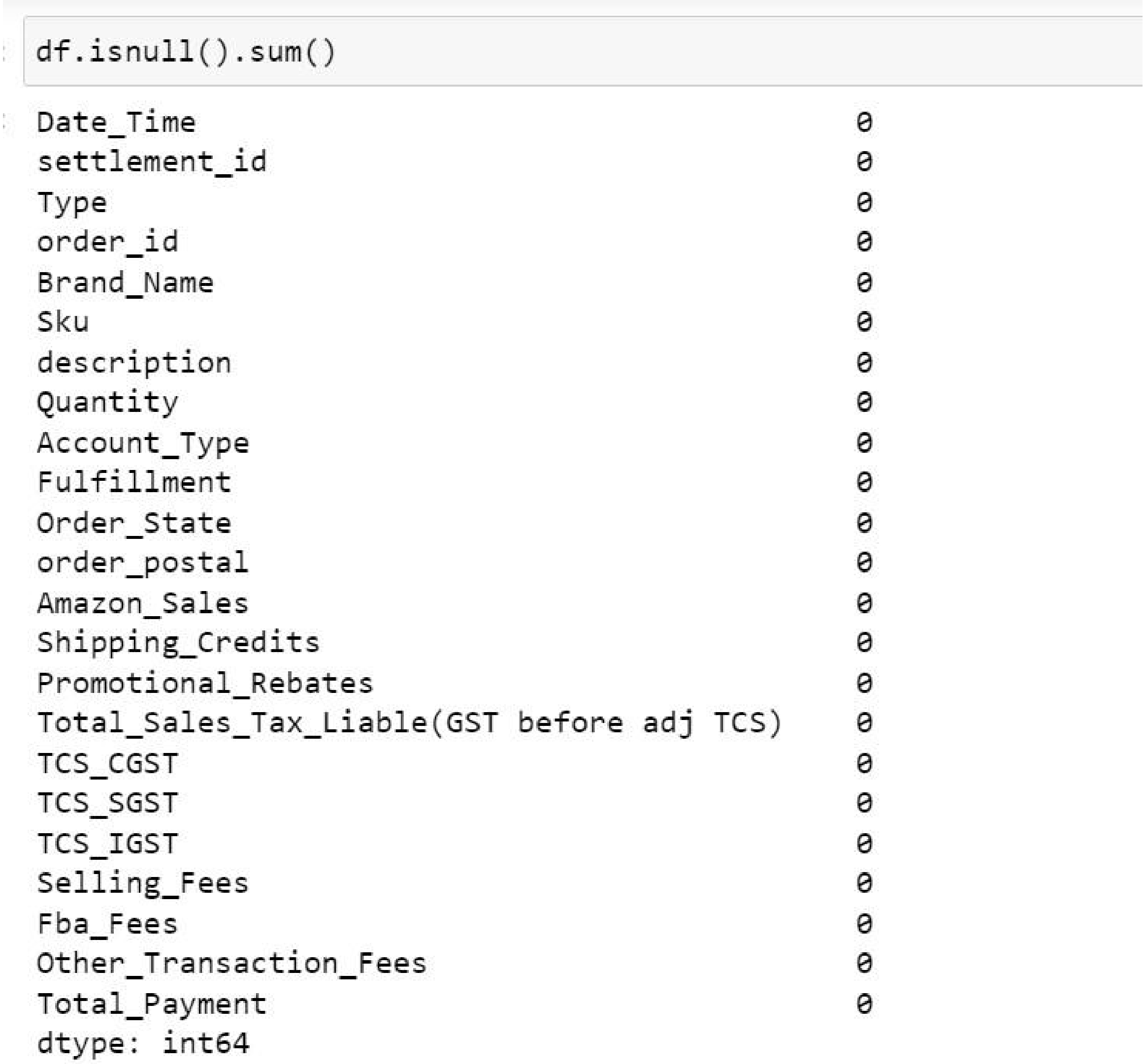
    Null Columns – 0

- **Target Variable:**

    Numerical Column – 1

DATA CLEANING:

In dataset null value treatment has been done.



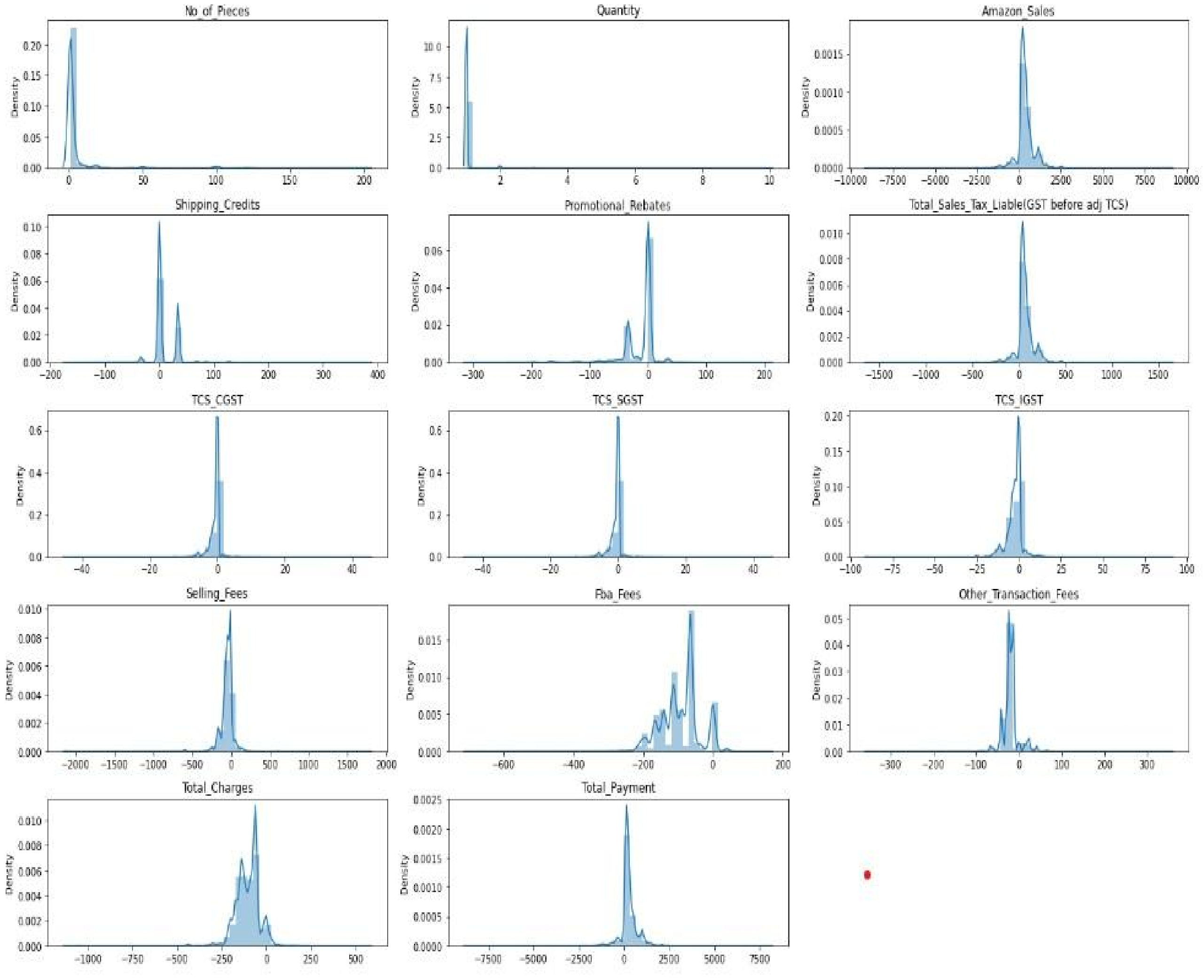
```
df.isnull().sum()
```

```
Date_Time                                    0
settlement_id                                0
Type                                         0
order_id                                     0
Brand_Name                                   0
Sku                                          0
description                                  0
Quantity                                     0
Account_Type                                 0
Fulfillment                                  0
Order_State                                  0
order_postal                                 0
Amazon_Sales                                 0
Shipping_Credits                             0
Promotional_Rebates                          0
Total_Sales_Tax_Liable(GST before adj TCS)   0
TCS_CGST                                     0
TCS_SGST                                     0
TCS_IGST                                     0
Selling_Fees                                 0
Fba_Fees                                     0
Other_Transaction_Fees                       0
Total_Payment                                0
dtype: int64
```

EXPLORATORY DATA ANALYSIS (EDA):

Relationship between the variables:

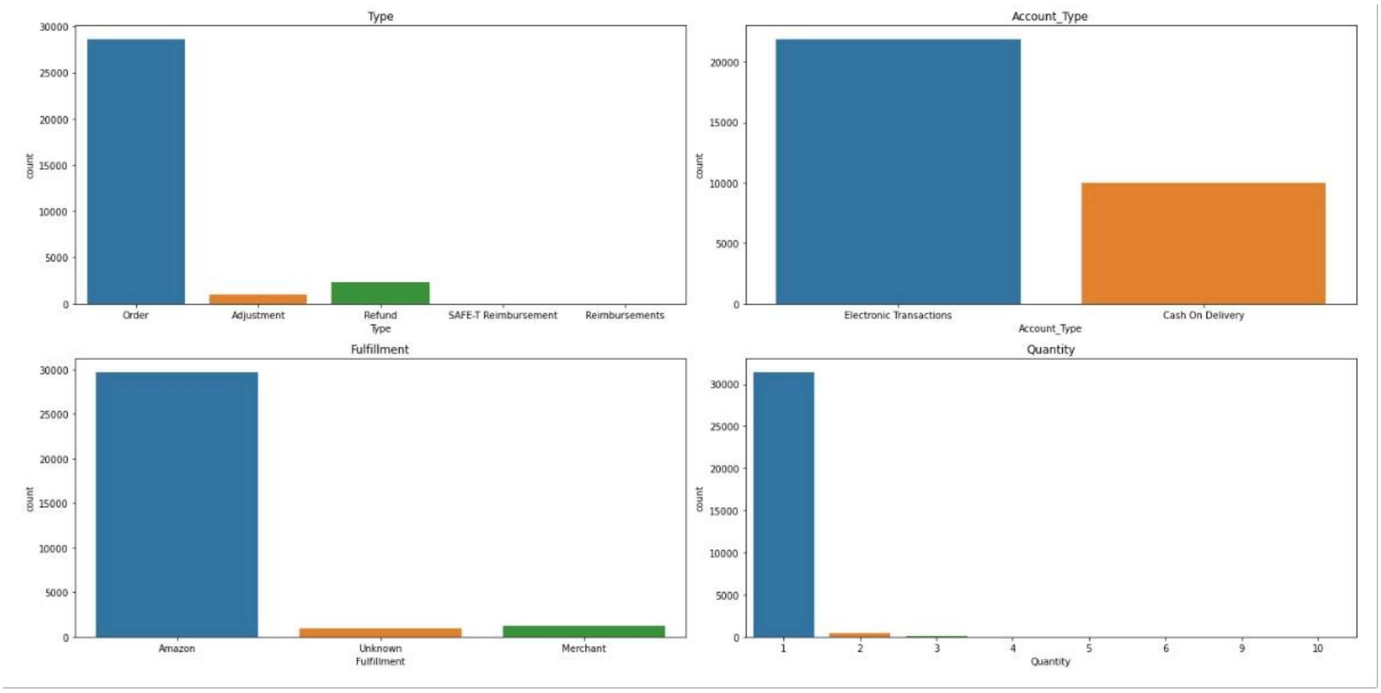
Uni-variate Analysis on Numerical Variables:



Inference:

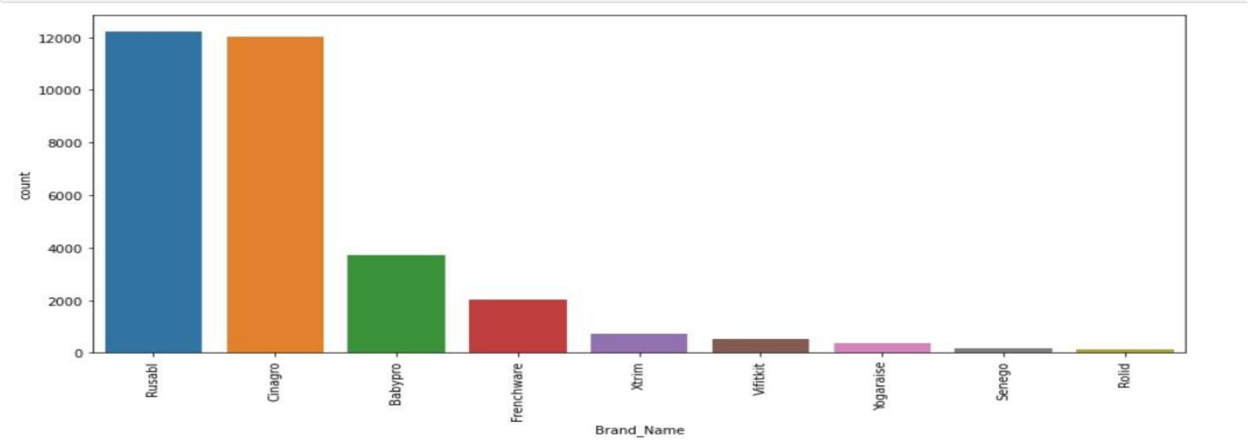
- Quantity is highly skewed towards right most of the orders ordered are 1 quantity.
- These below columns are symmetrically skewed Amazon sales, Shipping Credits, Promotional Rebates, Total Sales Tax Liable (GST before adjusting TCS), TCS-CGST, TCS-SGST, TCS-IGST, Fba Fees, Other Transaction Fees, Total Payment.
- The Selling Fees is slightly left skewed

Uni-variate Analysis on Categorical Variables:



Inferences:

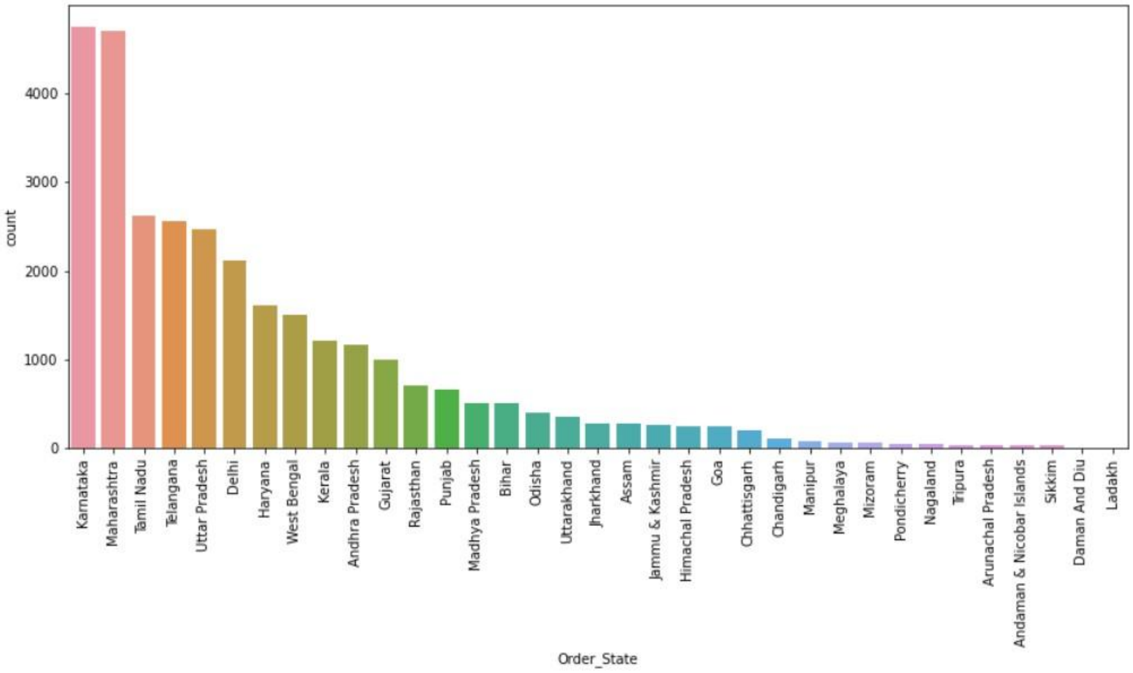
- Rusabl and Cinagro Products are high number of selling products.
- Electronic Transactions are more preferable.
- Most of the Sales are done by Amazon.



Inferences:

- The Products Rusabl and Cinagro have high number of sales.
- The Products Senego and Rolid has low number of sales.

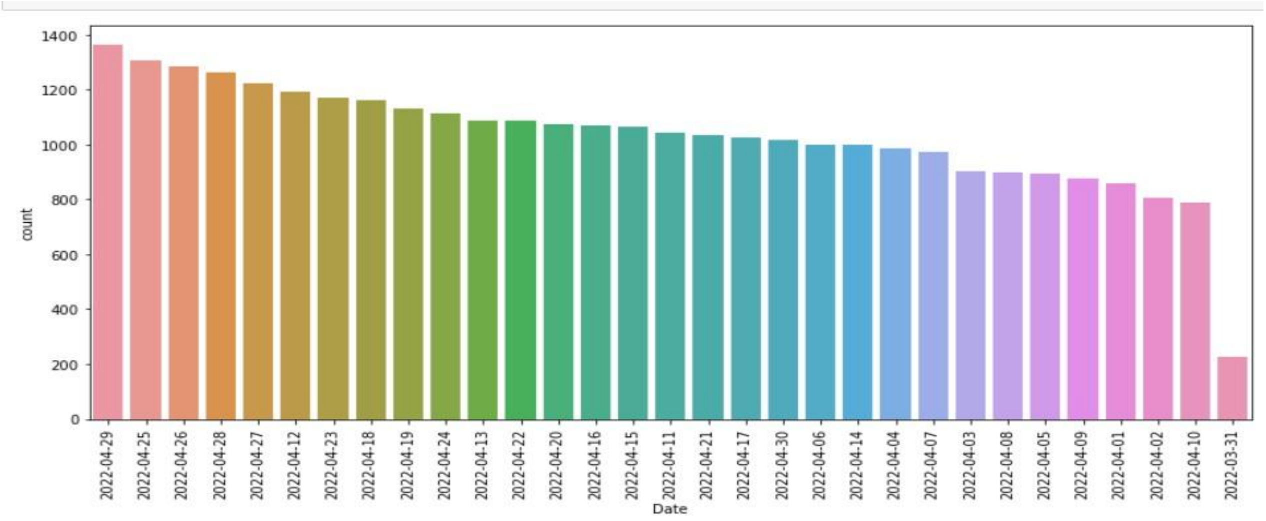
No of Orders for each State:



Inferences:

- Most of the Sales happened in Karnataka and Maharashtra
- The States Daman & Diu and Ladakh are having low sales

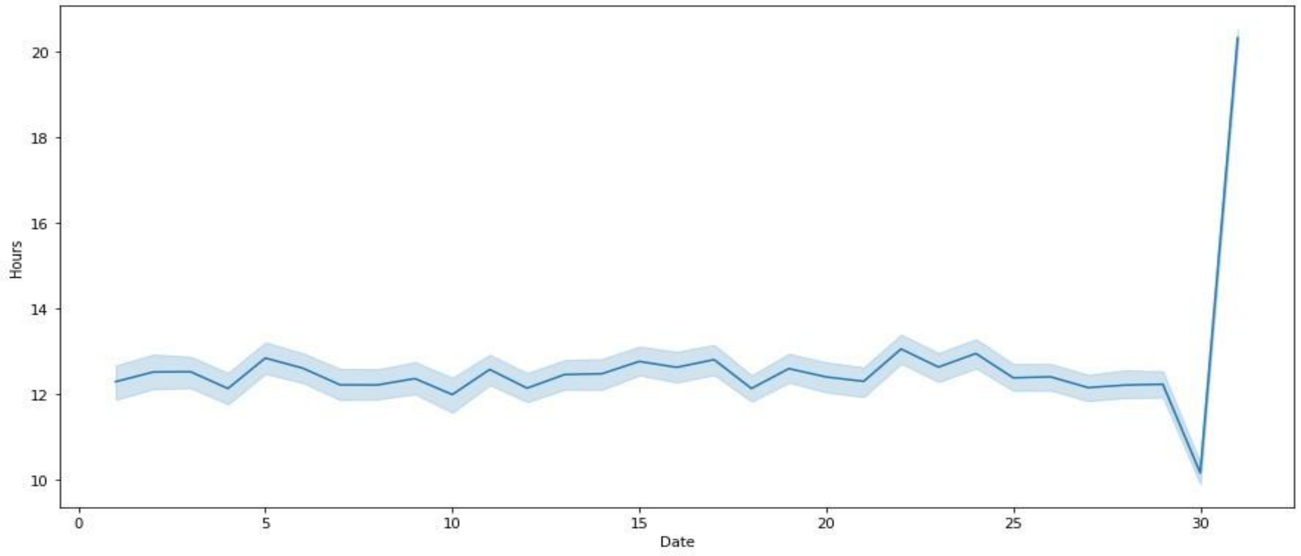
No of Orders placed for each Day:



Inferences:

- 29th of April 2022 has high sales and 31st of March 2022 has low sales compared to other days.

Hourly trend analysis for each day:



Inferences:

- Most of the orders take place during the Noon hours of the Day

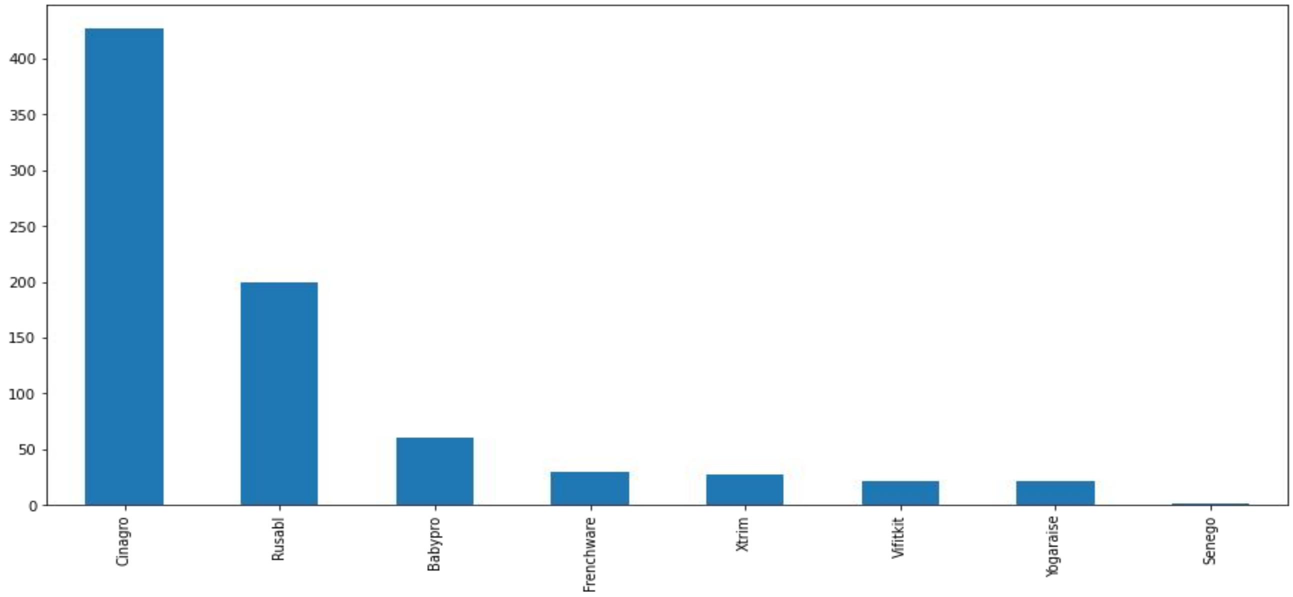
Analysis of records where Order State and Order Postal has Null Values:

We pulled out the records where we had order state and order postal null values and checked for the type and found out that all the records had a unique type of Adjustment. We assigned those records to Adj_orders and did some analysis on those records.

We are having the null values of order postal and order state because those are transaction are of

- FBA Inventory Reimbursement - Customer Return
- FBA Inventory Reimbursement - Customer Service Issue
- Reimbursement issued for lost or damaged package
- SAFE-T Claim on Amazon is Seller Assurance for Ecommerce Transactions. The essence of the program is that Amazon sellers can issue a refund to a customer in some cases. Namely, if: the losses occurred through no fault of the seller. the damage was during its preparation or delivery

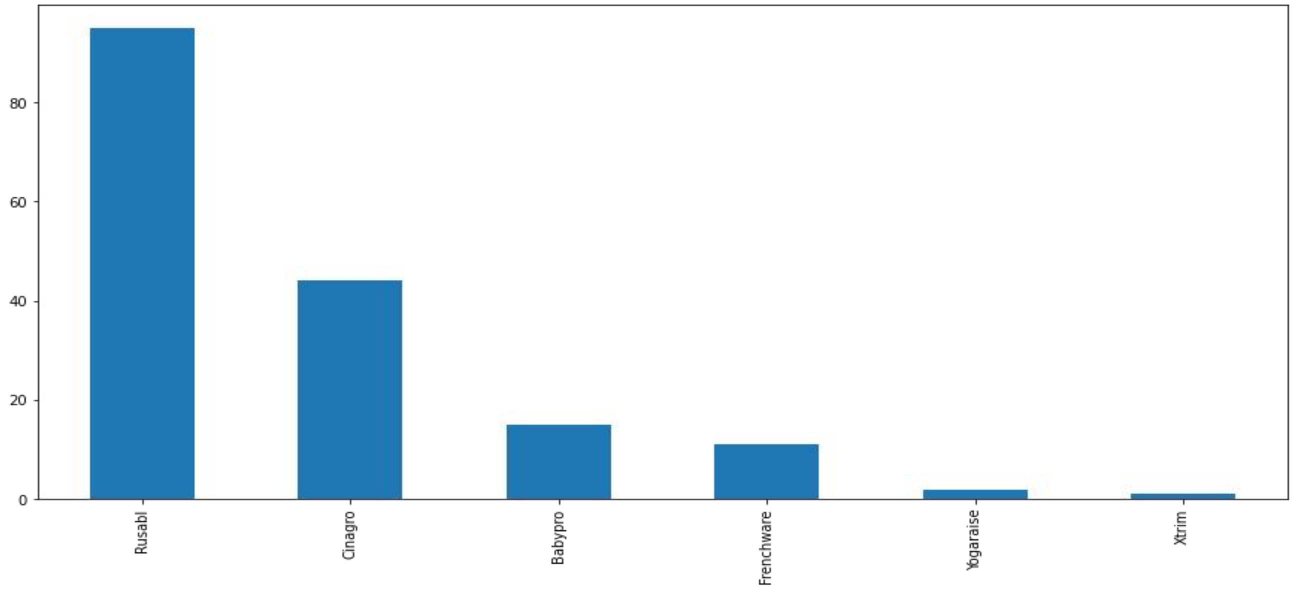
Plotted the no of orders returned due Customer Return for each brand:



Inferences:

- Except Rolid all other brands are getting customer return whereas Cinagro has highest no of returns.

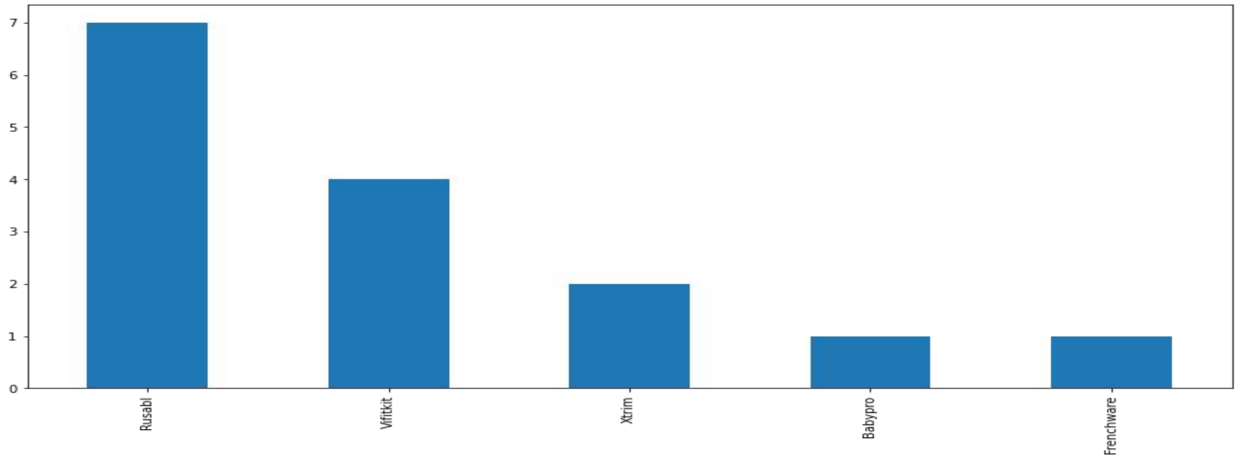
Plotted of the no of orders returned due Customer Service Issue for each brand:



Inferences:

- Rusabl has high customer service issues than all other brands.
- Rolid, Vifitkit, Senego has no customer service issue.

Plotted the no of orders returned due to Reimbursement issued for lost or damaged package for each brand:

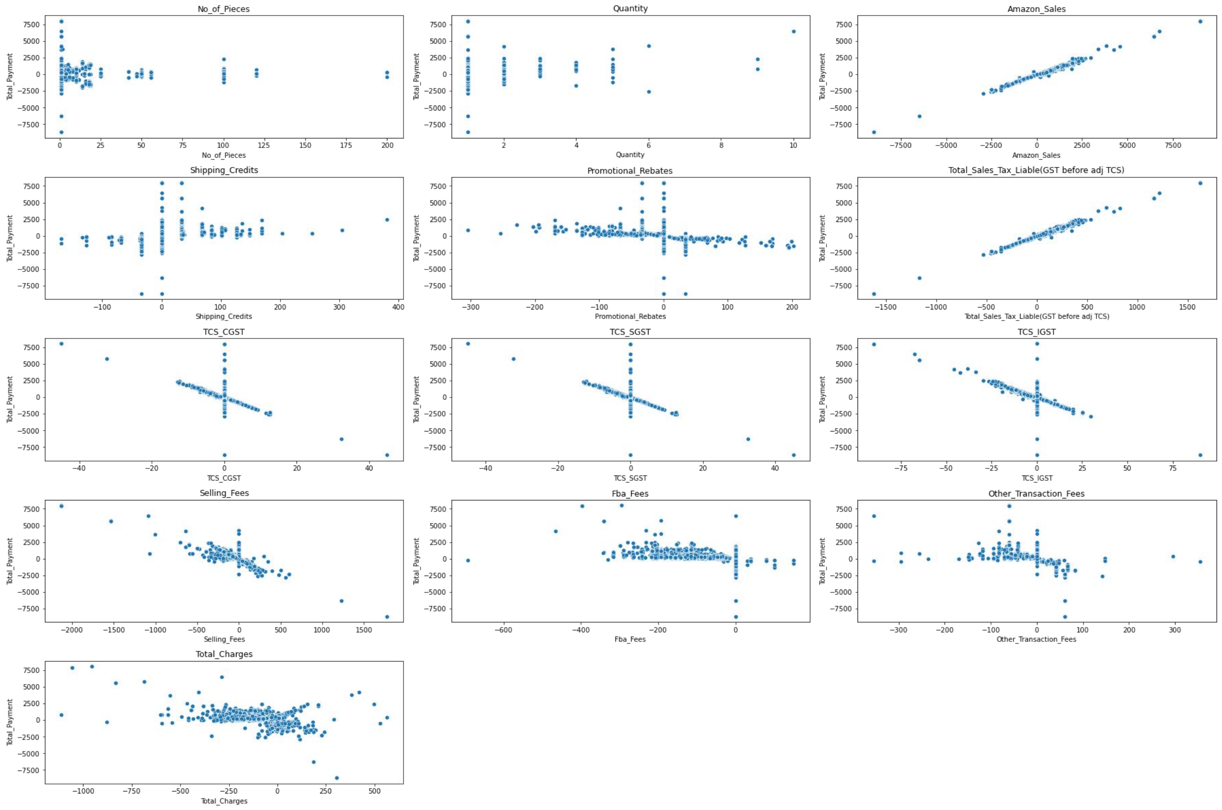


Inferences:

Rusabl has the highest number of lost or damaged package delivery followed by Vifitkit.

For Further Analysis, we dropped the null values because as the orders were not successfully delivered to the customers, so we are dropping these records before model building.

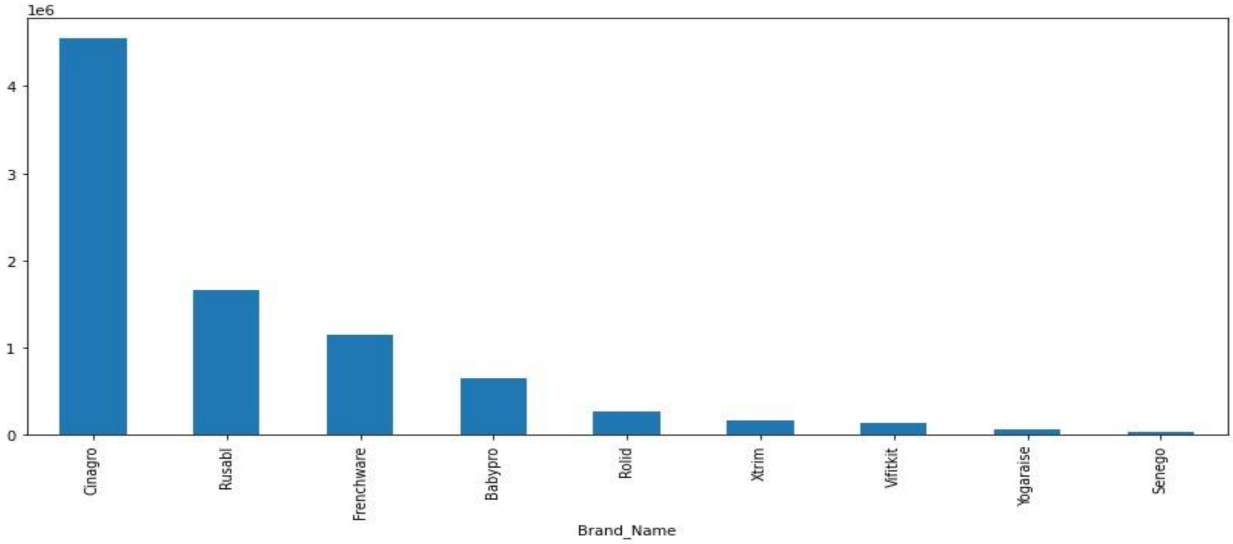
Bivariate Analysis of Numerical Variable:



Inferences:

- Total_Payment and 'Amazon Sales', 'Total_Sales_Tax_Liable (GST before adj TCS)' has linear relationship with each other.
- TCS_CGST, TCS_SGST, TCS_IGST and Total_Payment has negative linear relationship with each other.

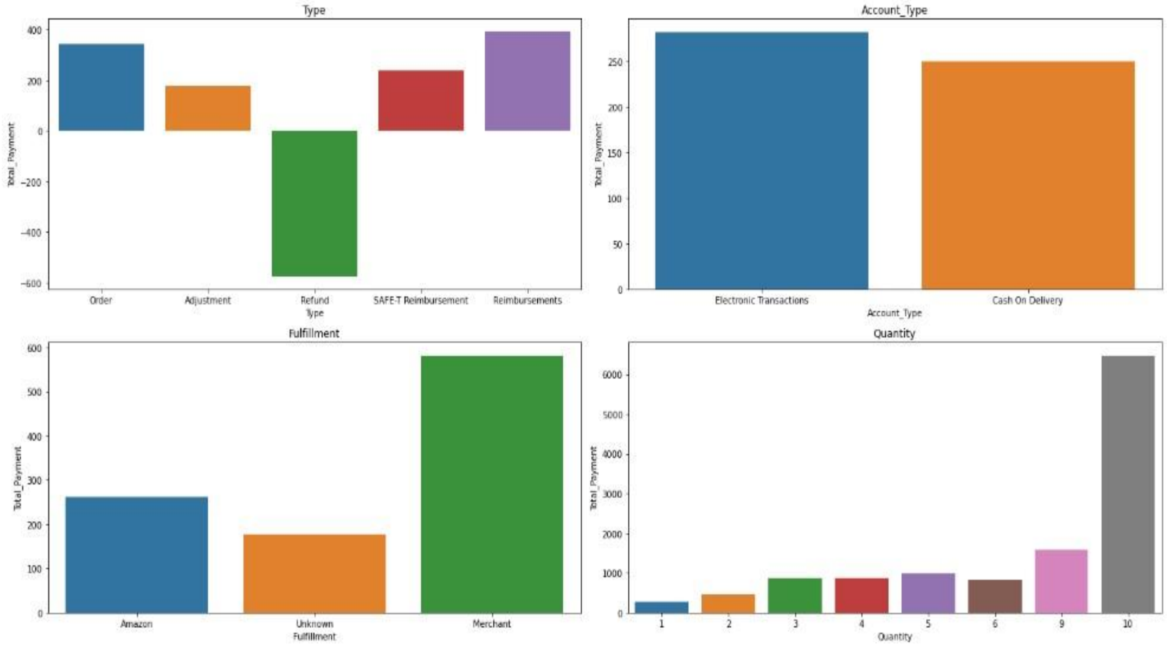
Total Payment for Each Brand:



Inferences:

- The Brand Cinagro has highest sales around 45L.
- The Brand Senego has lowest sales around 39K.

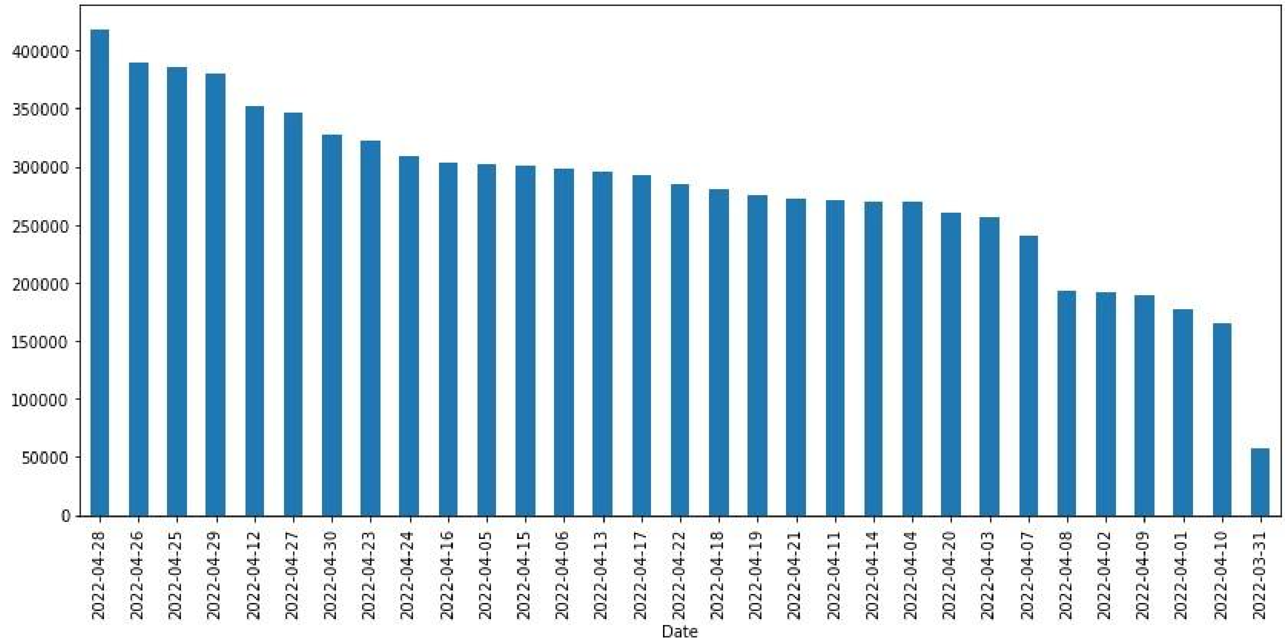
Bivariate Analysis on Categorical Variables:



Inferences:

- The Total_Payment for Type Reimbursements are highly spent and the Refund type has negative spent of Total_Payments because those payments are refunded from the amazon to the customers.
- Electric Transactions have high Total_Payment amount preferred by the customers compared to Cash on Delivery.
- Merchant got the highest amount of Total_Payment.
- The Quantity of 10 has highest Total_Payment.

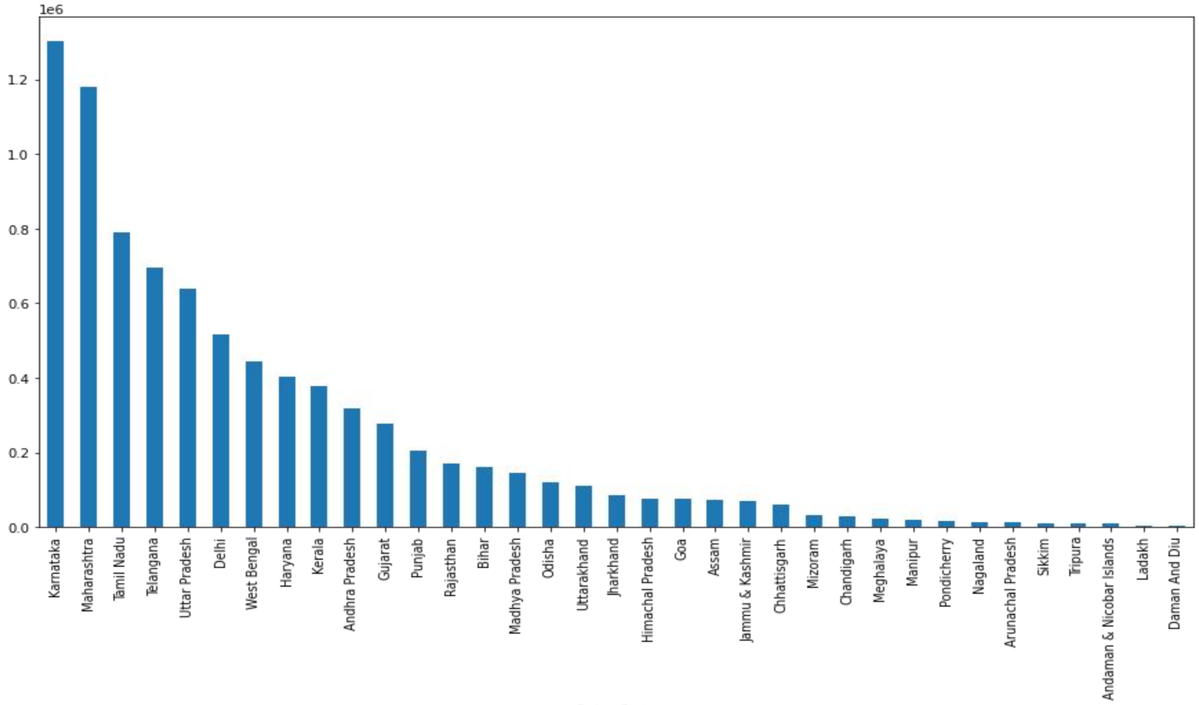
Total Payment for each Day:



Inferences:

- 28th of April 2022 has the highest sales around 4 lakhs.
- 31st of March 2022 has the lowest sales around 50K.

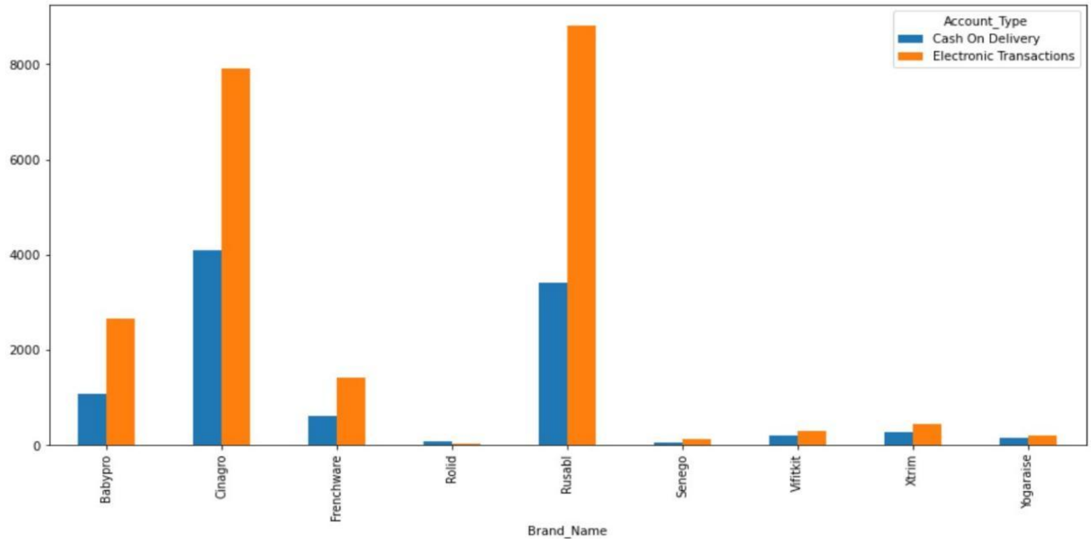
Total Payment for each State:



Inferences:

- The State Karnataka has highest sales around 13L.
- The State Daman and Diu has lowest sales around 3K.

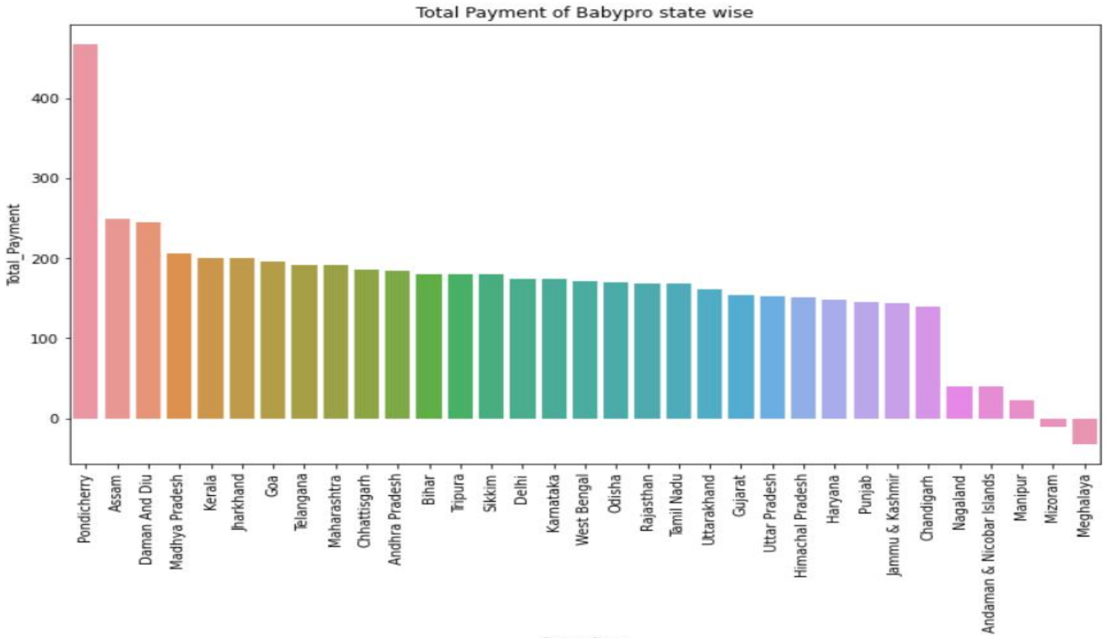
Account Type preferred for Payment for each Brand:



Inferences:

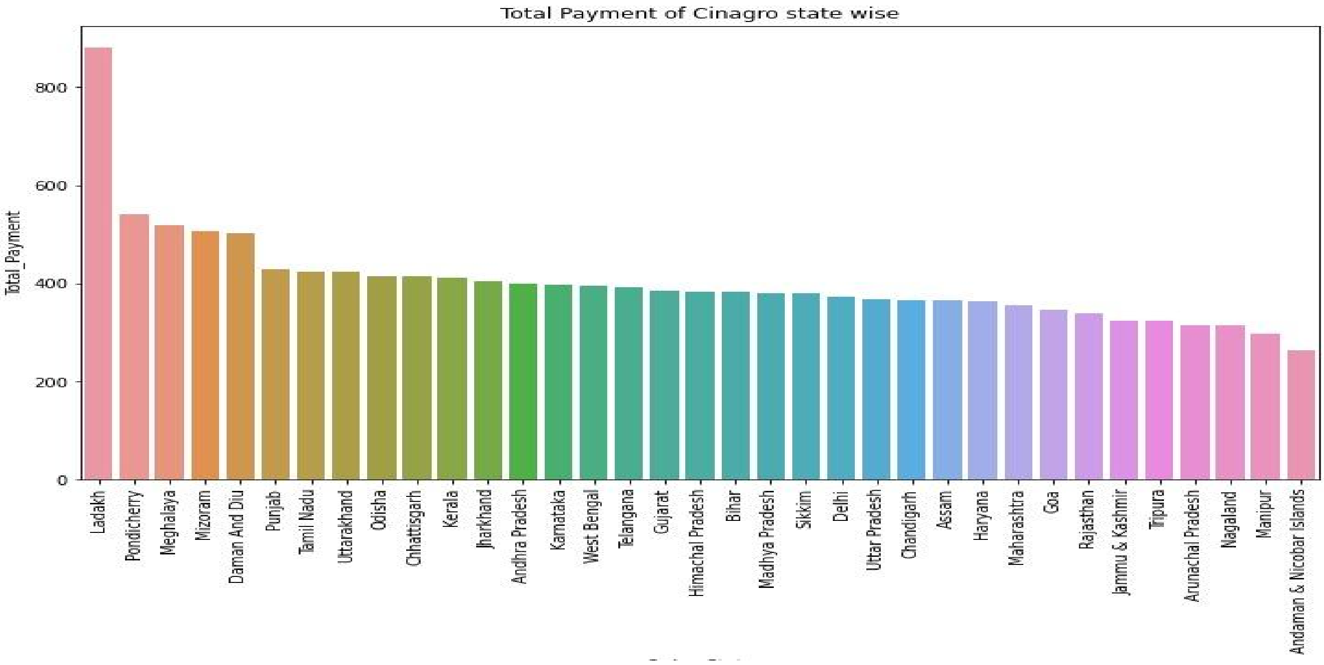
- Rolid Brand customers prefer cash on delivery option than Electronic Transactions since rolid Brand sells Metallic Locker products and it is costly compared to other brand products.
- Other Brands Prefers Cash on Delivery.

Total Payment of each Brand with respect to Order State:



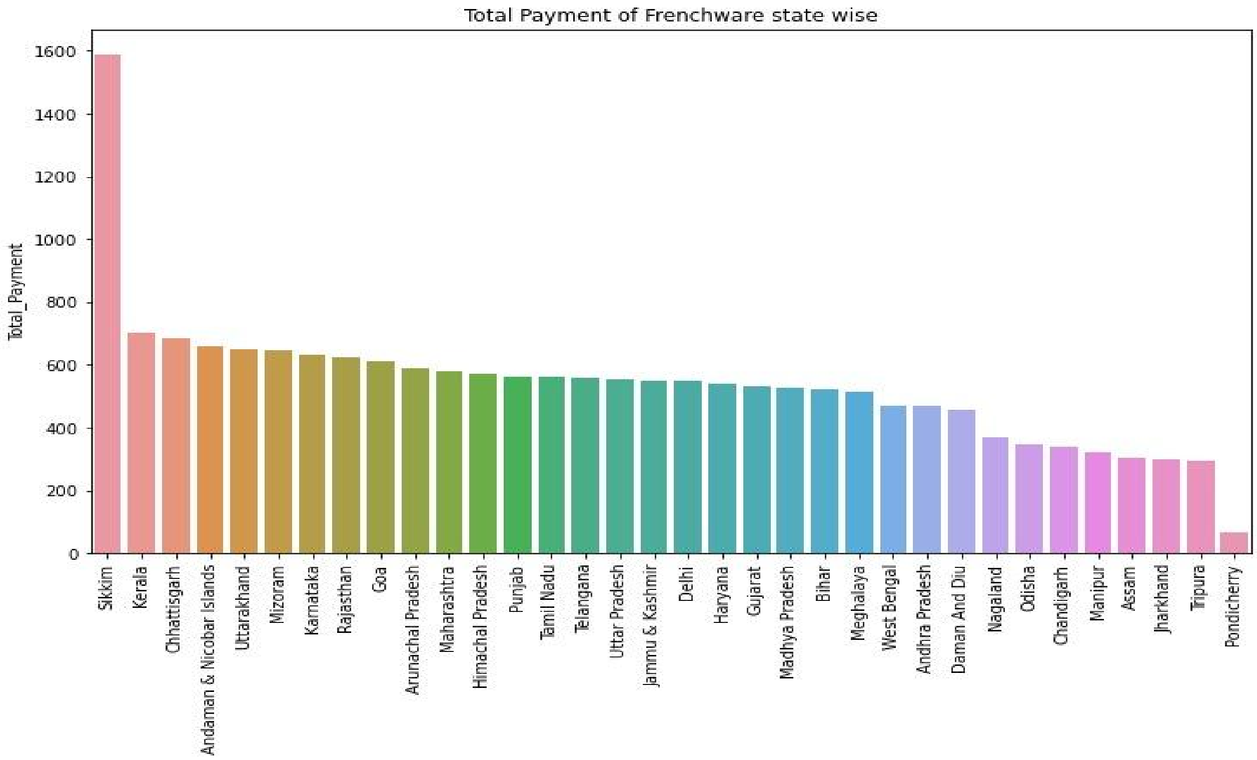
Inferences:

- Most of Total Payment of BabyPro are from Pondicherry and the states Mizoram, Meghalaya has negative sales since it has refund.
-



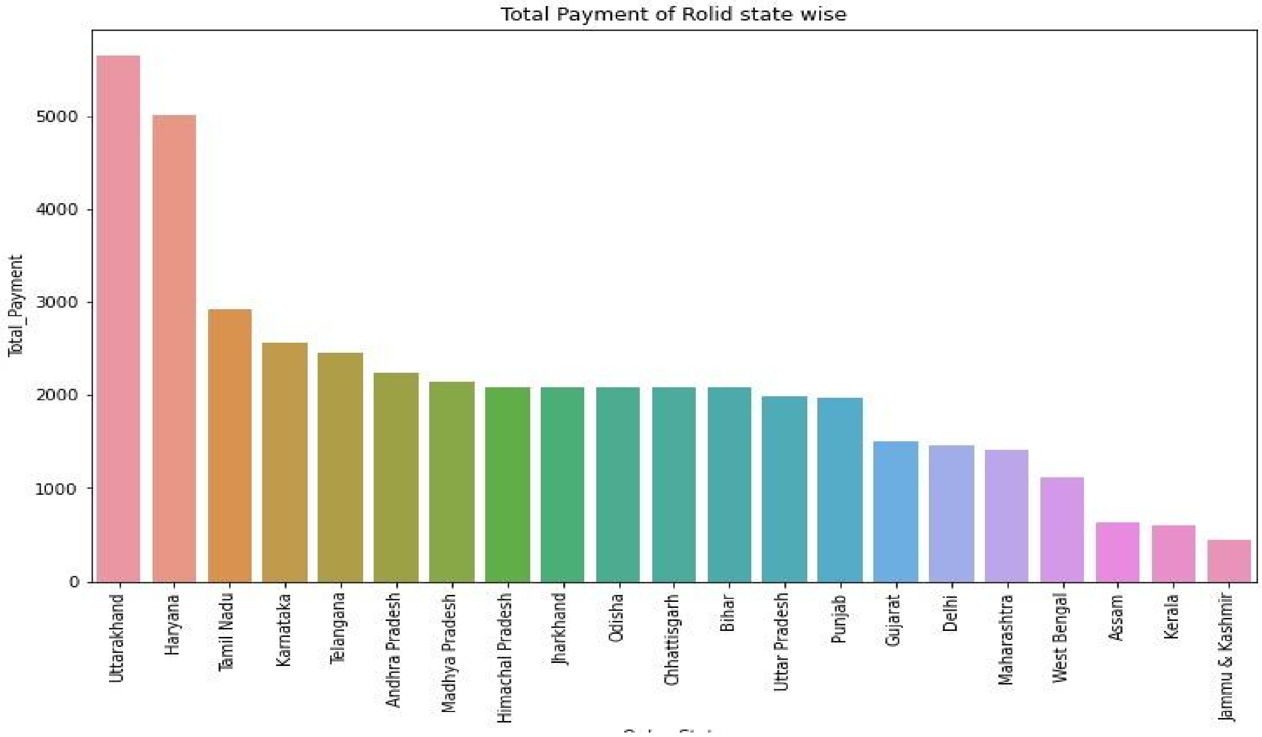
Inferences:

- Most of Total Payment of Cinagro are from Delhi and Manipur has low sales.



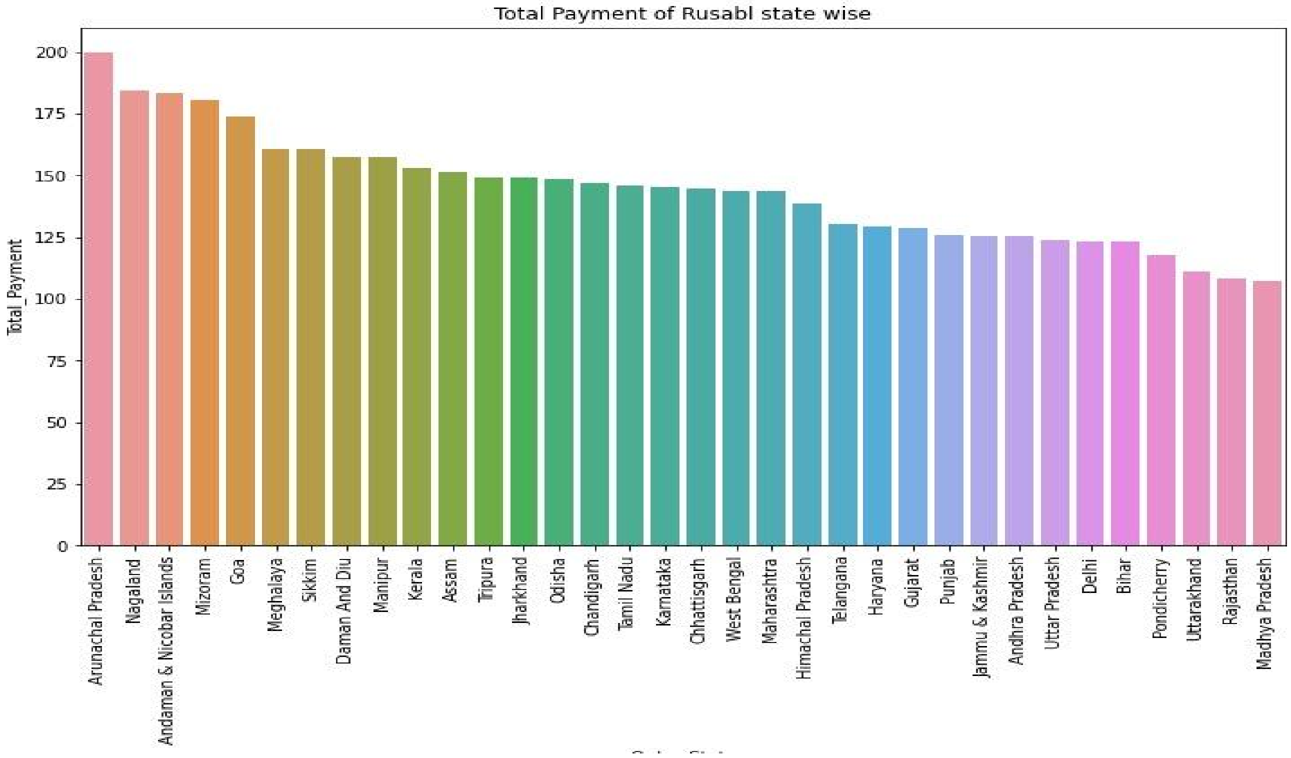
Inferences:

- Most of Total Payment of Frenchware are from Sikkim and Pondicherry has low sales.



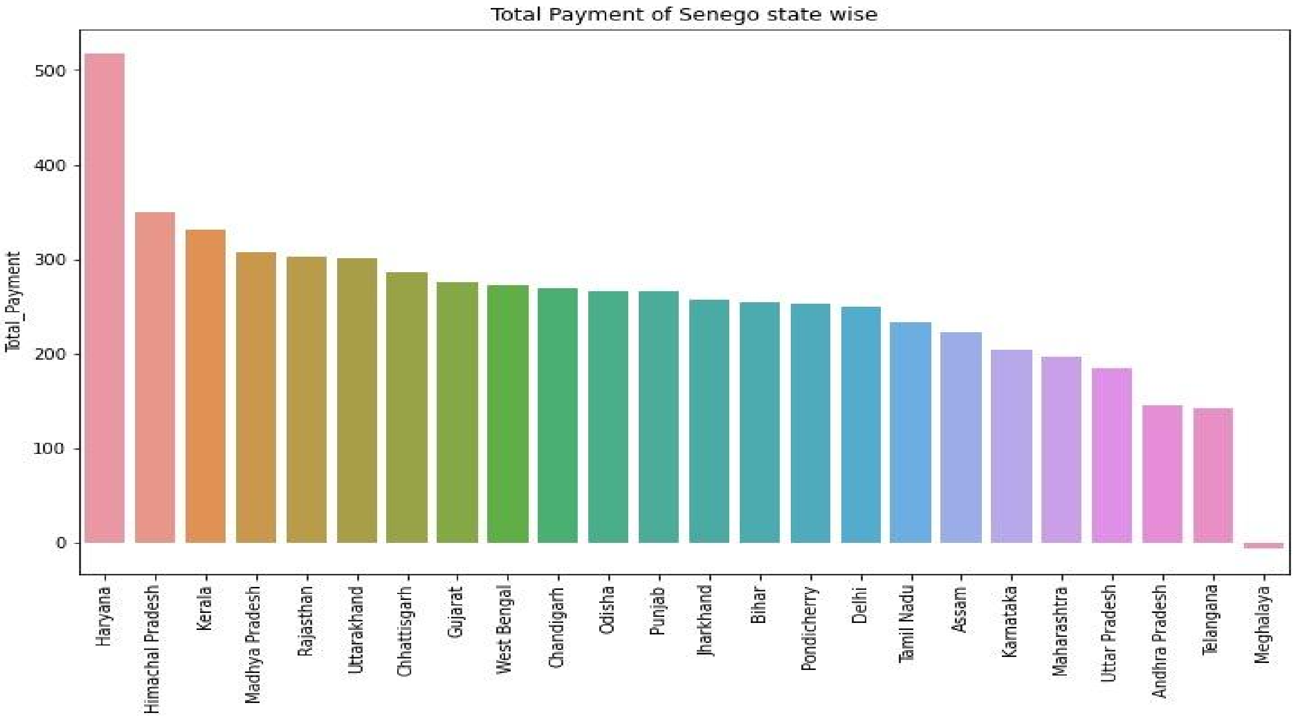
Inferences:

- Most of Total Payment of Rolid are from Uttarakhand and Jammu & Kashmir has low sales.



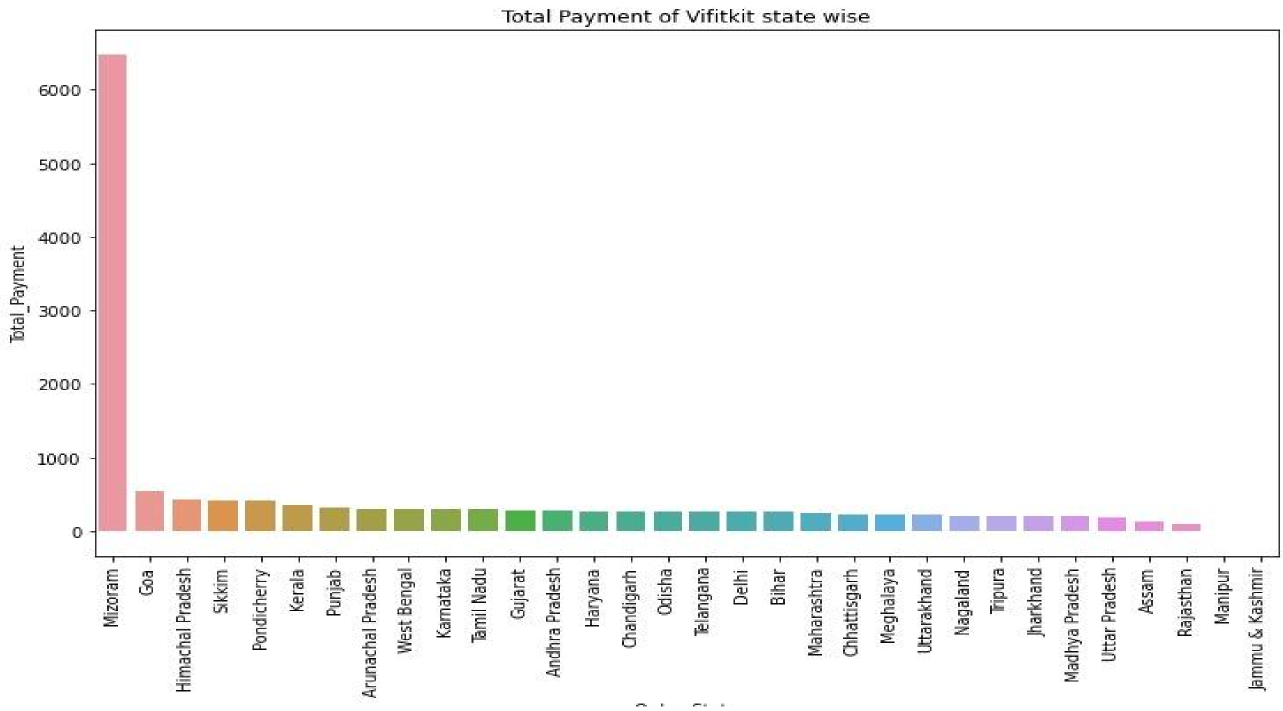
Inferences:

- Most of Total Payment of Rusabl are from Arunachal Pradesh.



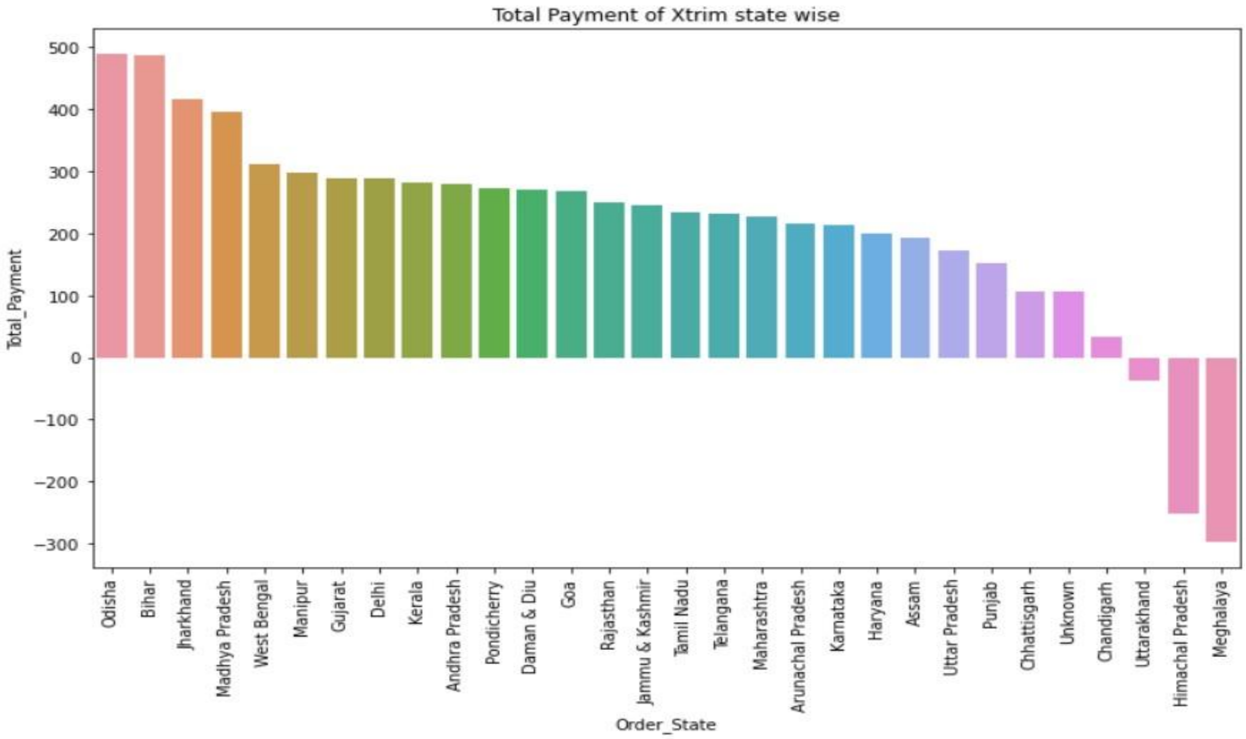
Inferences:

- Most of Total Payment of Senego are from Haryana and Meghalaya has low sales.



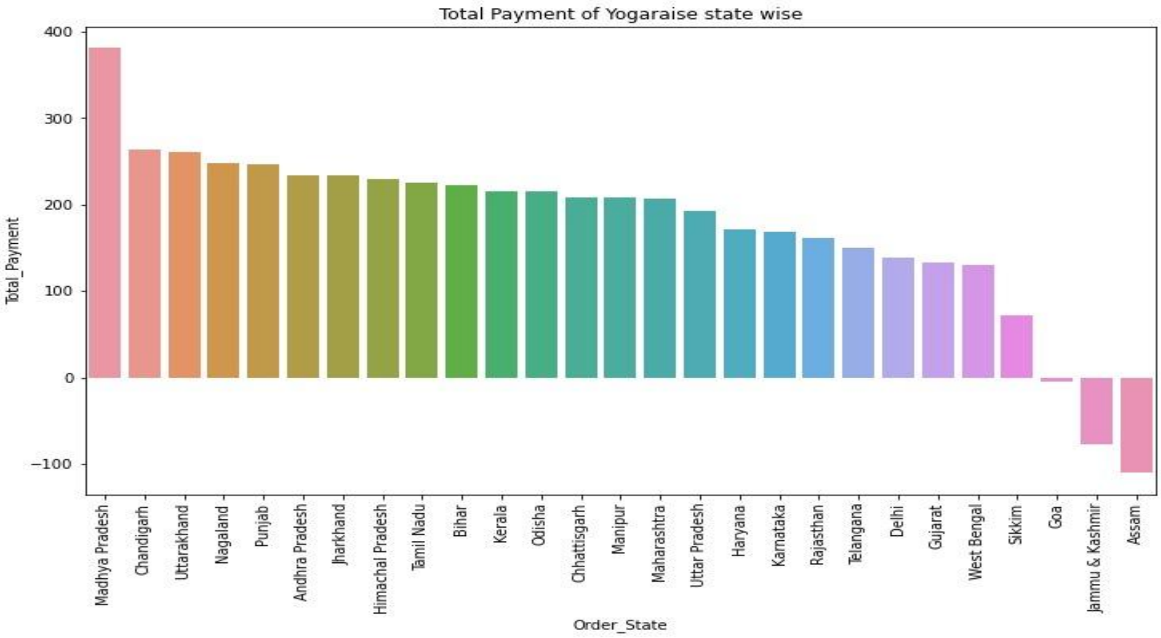
Inferences:

- Most of Total Payment of Vifitkit are from Mizoram.



Inferences:

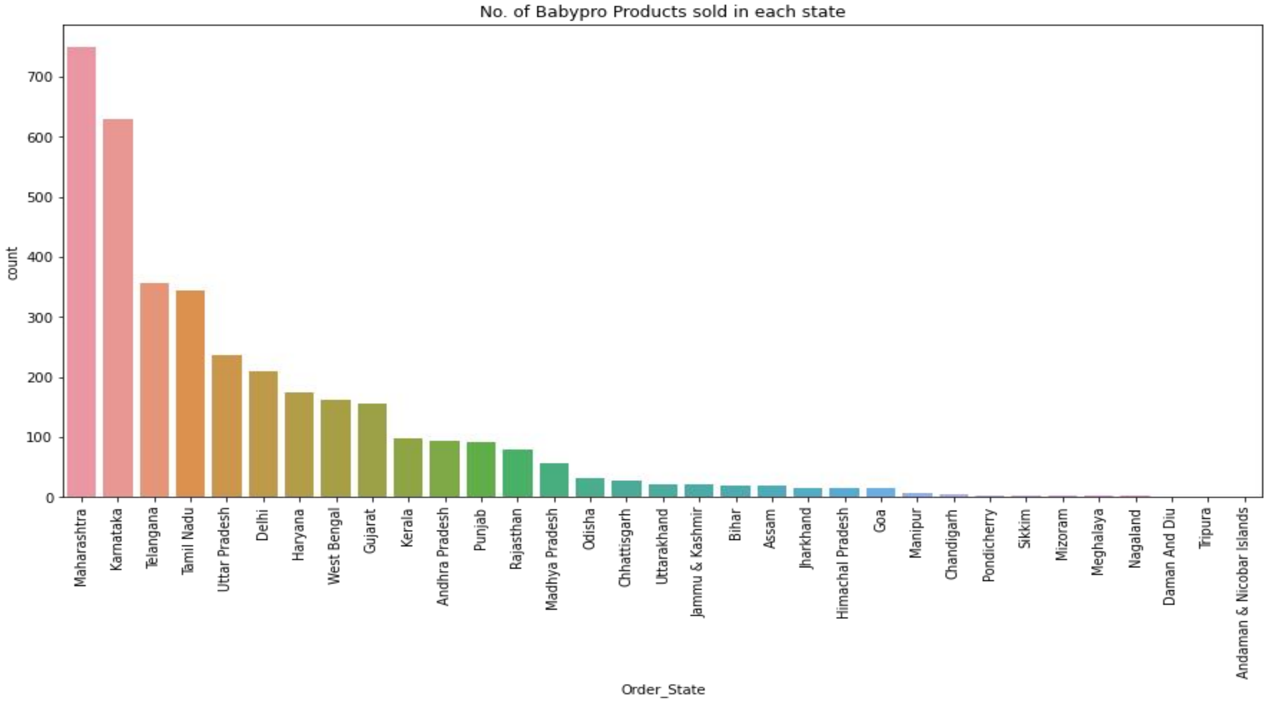
- Most of Total Payment of XTrim are from Odisha and the states Uttarakhand, Himachal Pradesh, Meghalaya has negative sales.



Inferences:

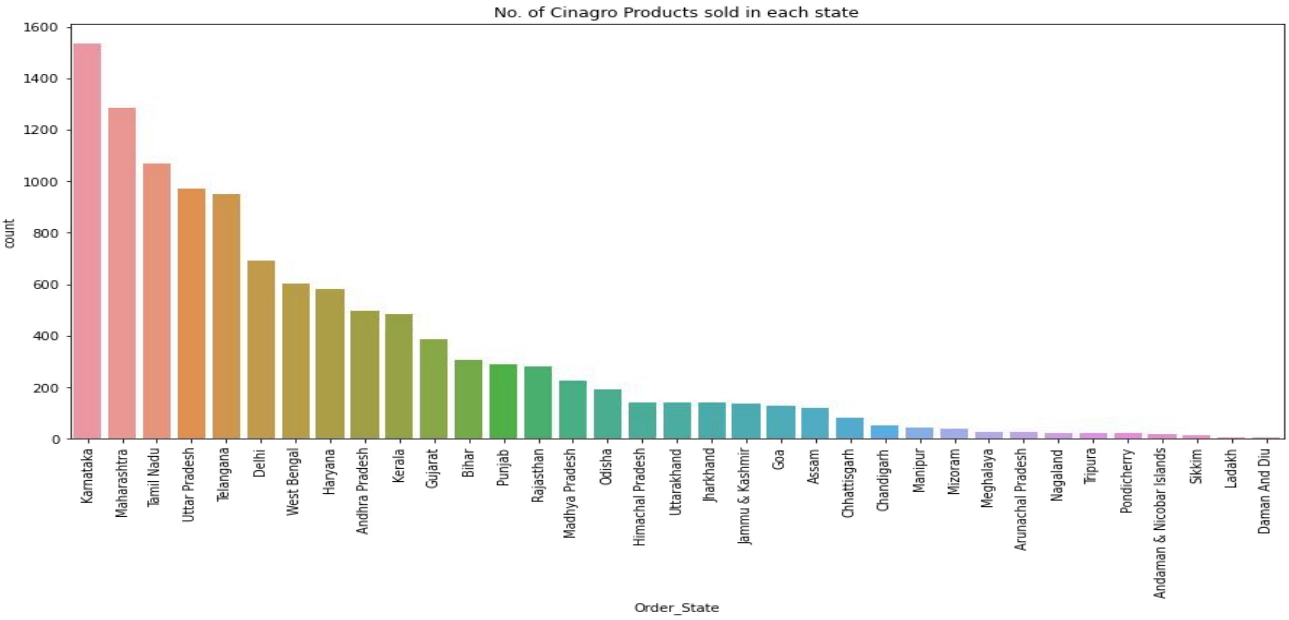
- Most of Total Payment of Yogaraise are from Madhya Pradesh and the states Jammu & Kashmir, Goa and Assam has negative sales.

No. of products sold in each State with respect to Brand wise:



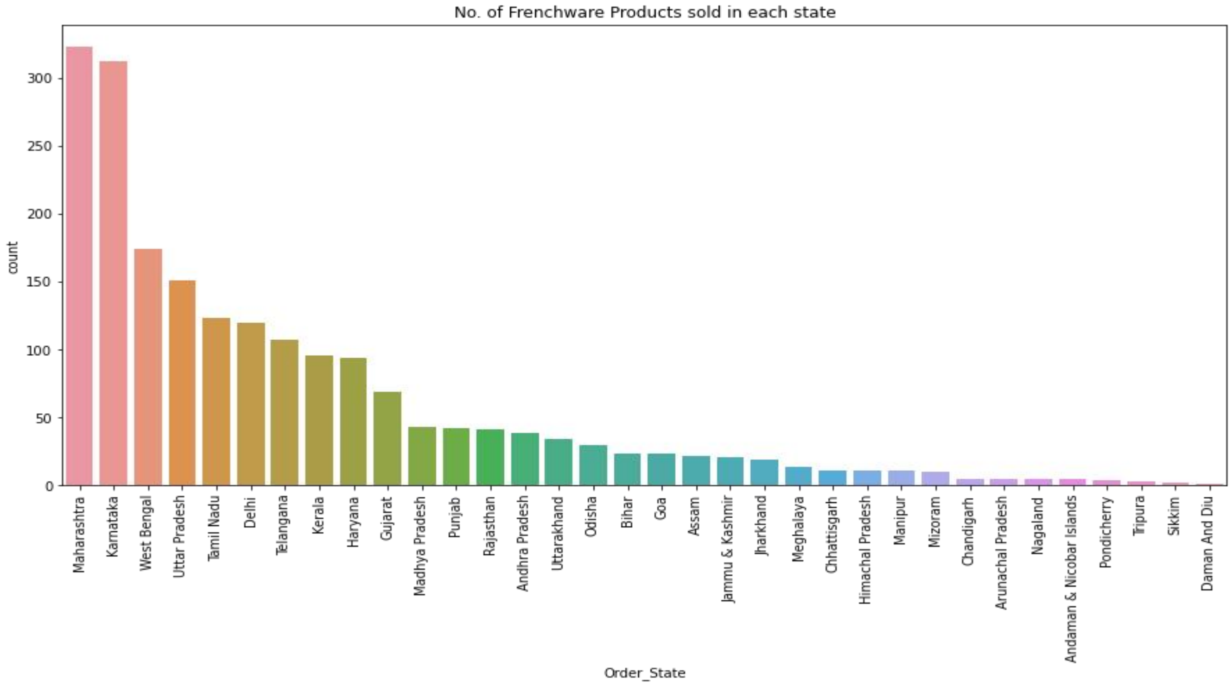
Inferences:

- The BabyPro has high sales in Maharashtra followed by Karnataka.



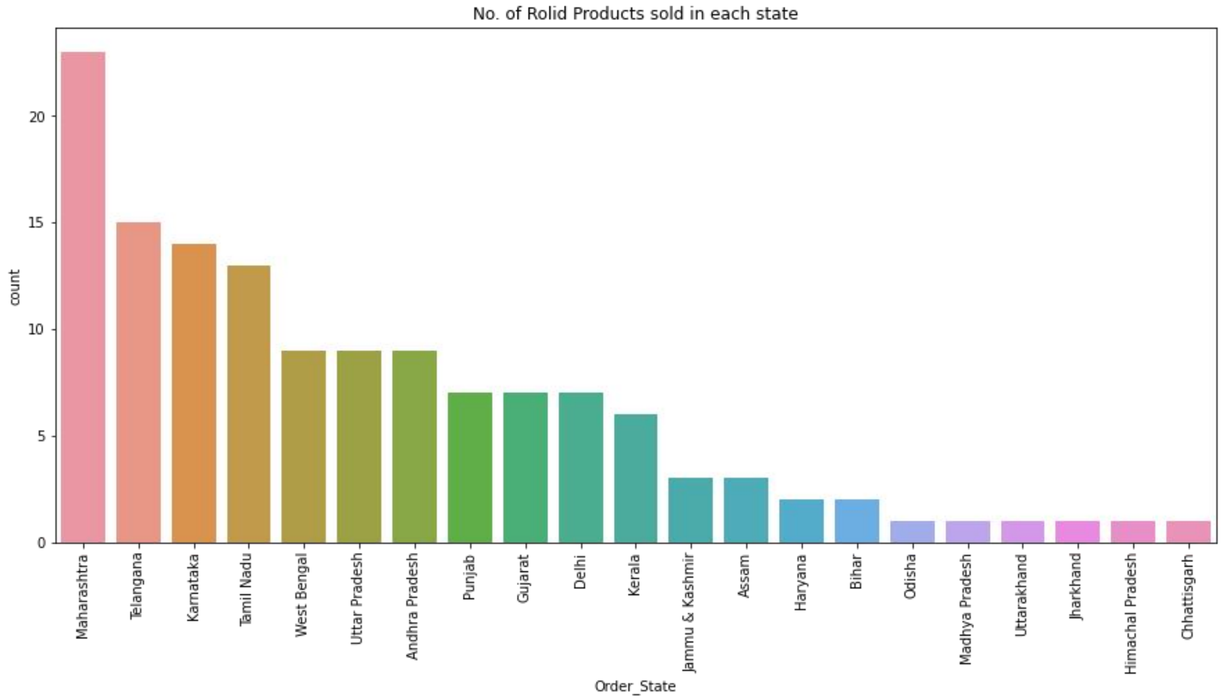
Inferences:

- The Cinagro has high sales in Karnataka followed by Maharashtra and Tamil Nadu.
- The Cinagro has low sales Daman and Diu and Ladakh.



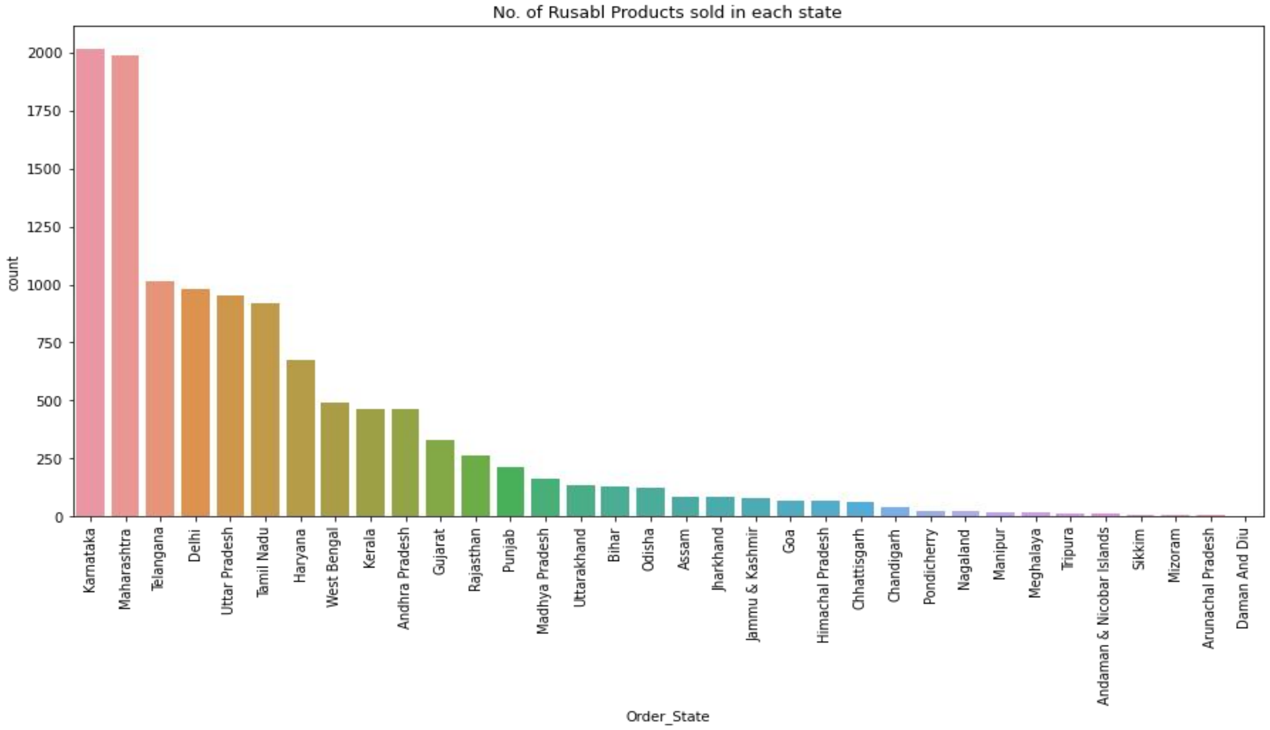
Inferences:

- The Frenchware has high sales in Maharashtra and Karnataka.



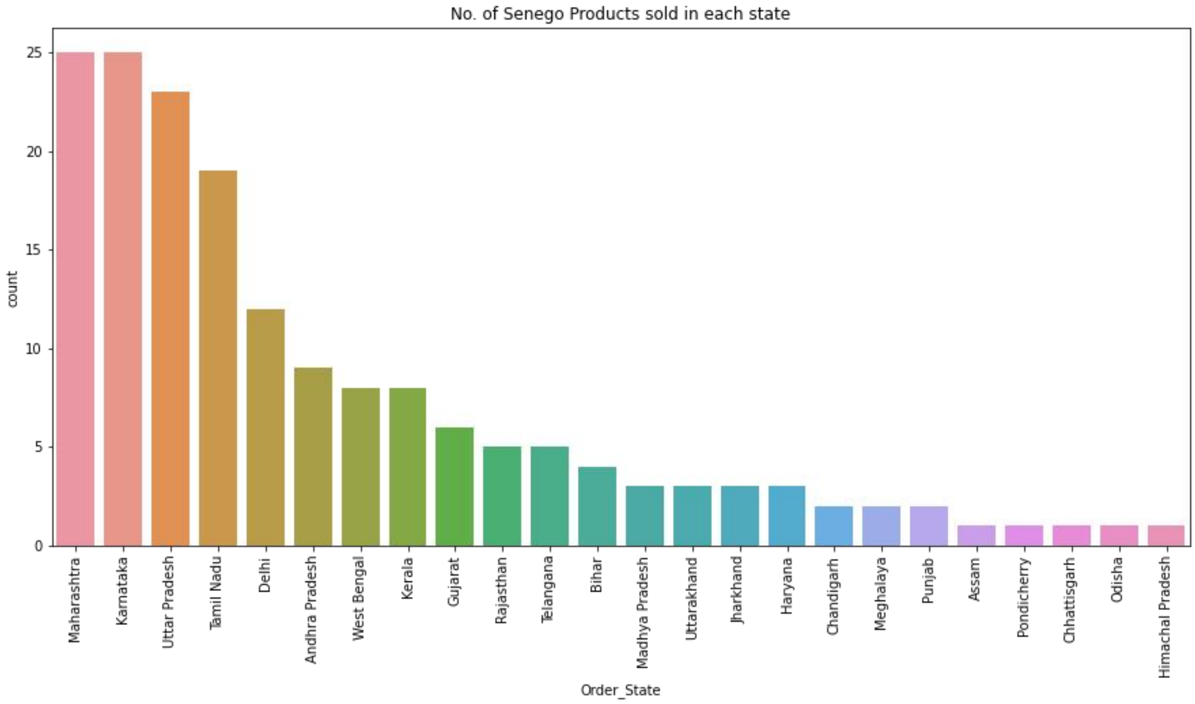
Inferences:

- The Rolid has high sales in Maharashtra followed by Telangana.



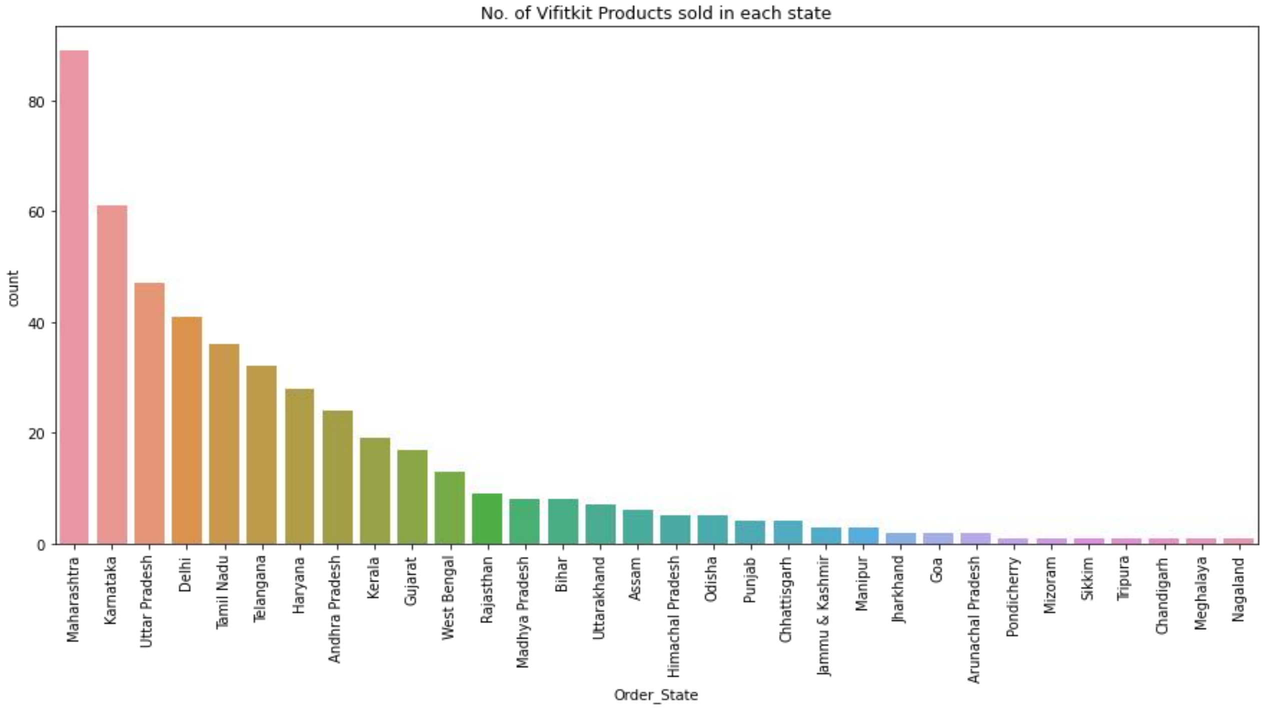
Inferences:

- The Rusabl has high sales in Karnataka and Maharashtra.



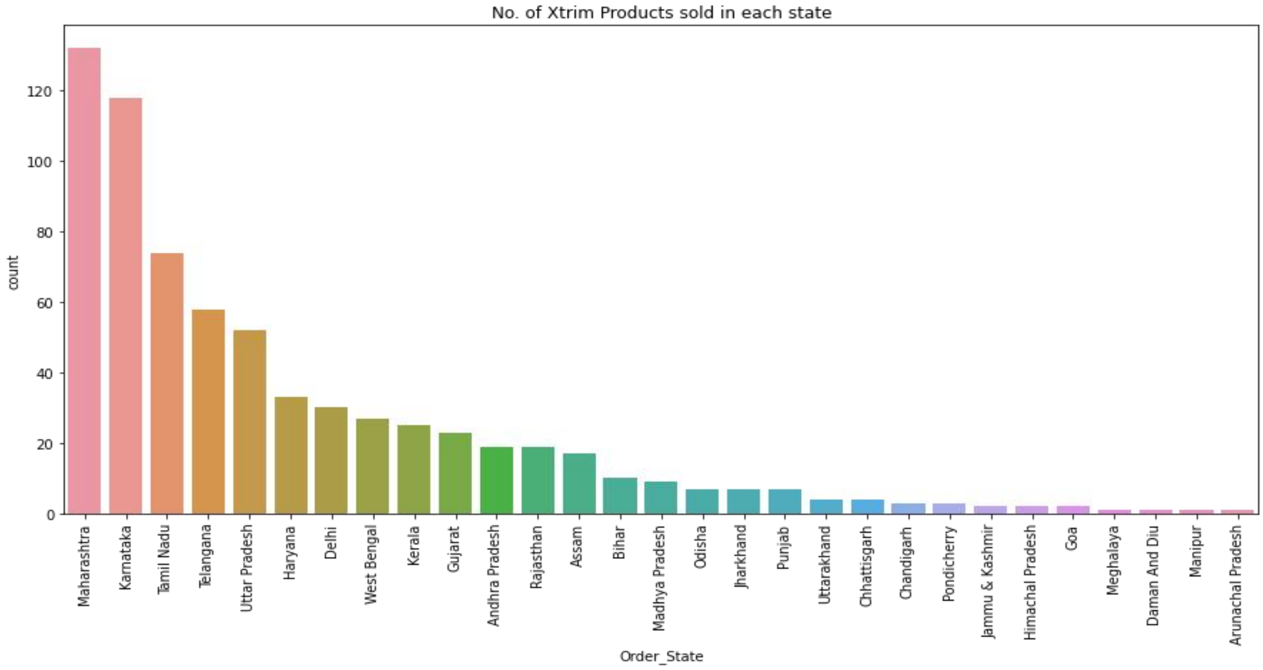
Inferences:

- The Senego product has high sales in Maharashtra and Karnataka.



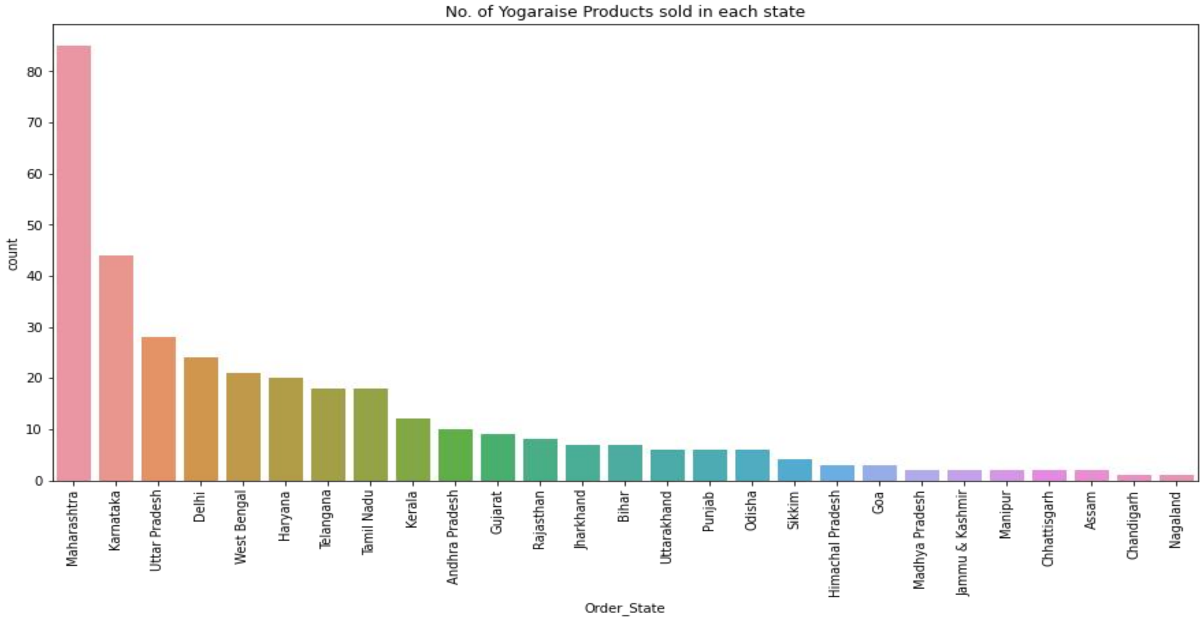
Inferences:

- The Vifitkit product has high sales in Maharashtra and low sales in Mizoram, Meghalaya, Chandigarh, Nagaland, Sikkim, Pondicherry, and Tripura.



Inferences:

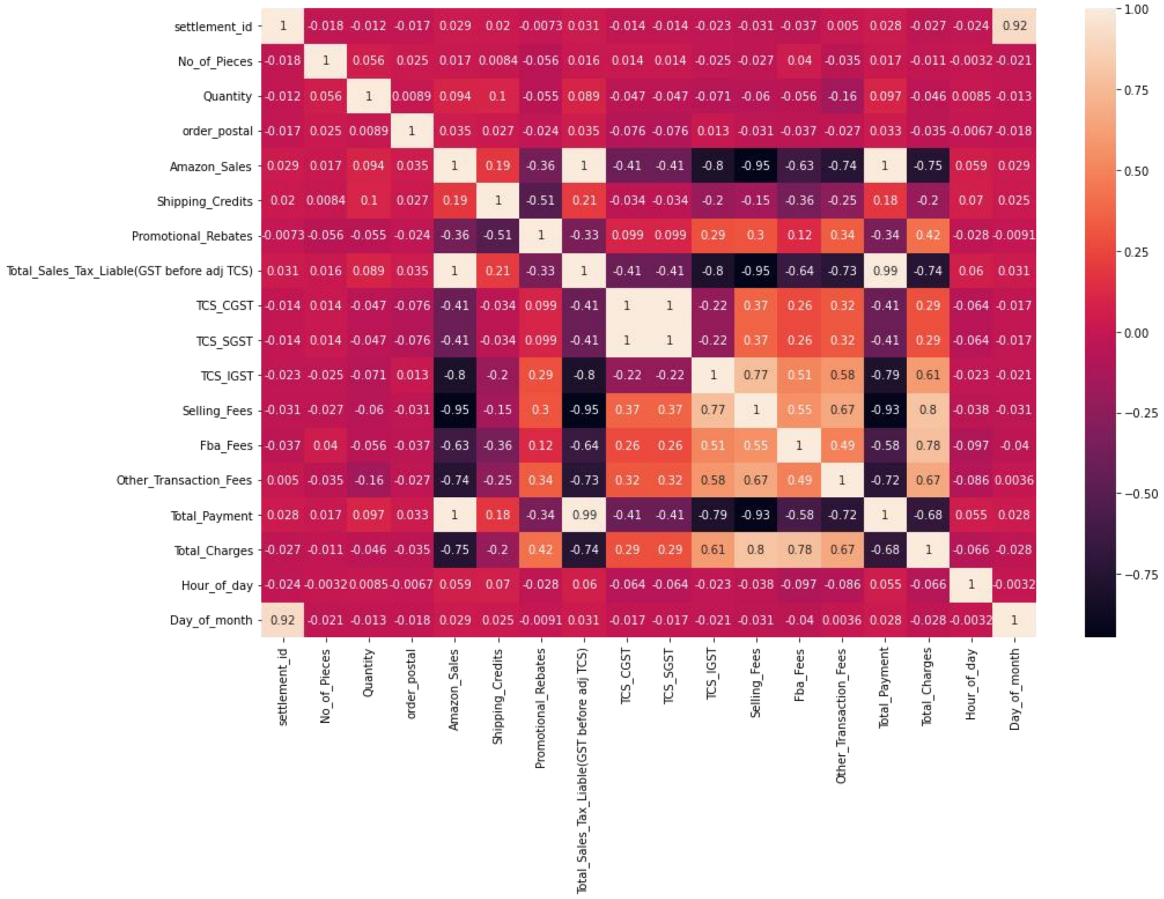
- The XTrim has high sales in Maharashtra and followed by Karnataka and Tamil Nadu.



Inferences:

- The Yogaraise has high sales in Maharashtra followed by Karnataka

Heatmap:

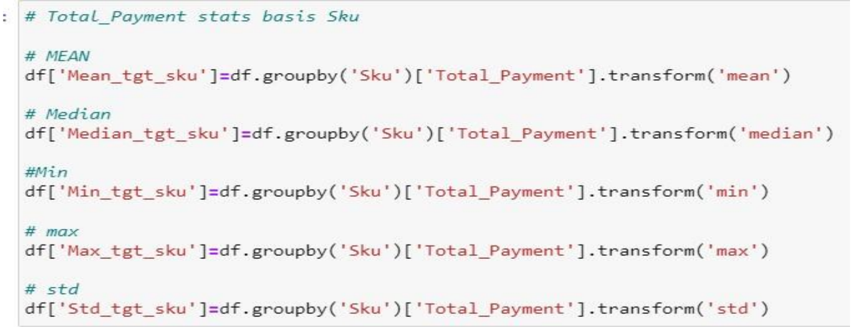


Inferences:

- The Target variable has high correlation between amazon sales, Total_Sales_Tax. So, before model building, we need to remove those columns
- The Target variable has negative correlation between GST columns, Selling Fees, FBA fees.

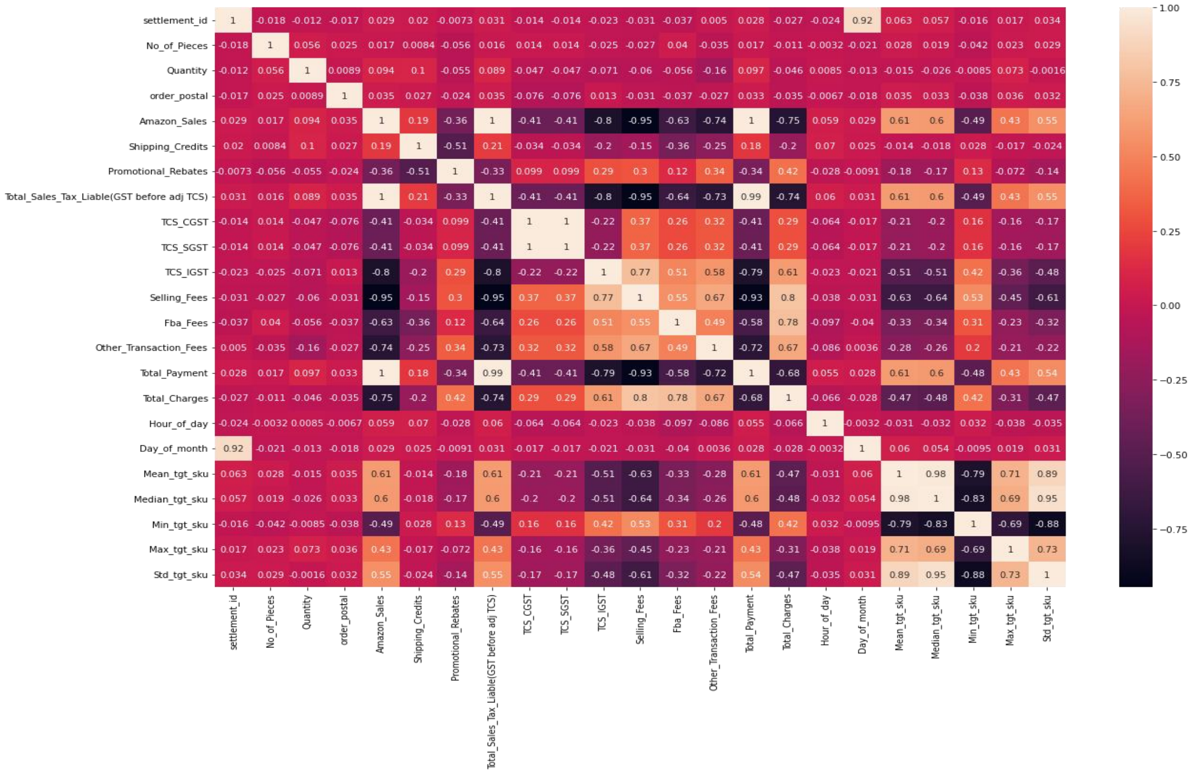
Creating Variables to Adjust the Correlation:

Created a new variable based on the Sku and transformed with five-point summary statistics and saved it in new columns.



```
# Total_Payment stats basis Sku

# MEAN
df['Mean_tgt_sku']=df.groupby('Sku')['Total_Payment'].transform('mean')

# Median
df['Median_tgt_sku']=df.groupby('Sku')['Total_Payment'].transform('median')

#Min
df['Min_tgt_sku']=df.groupby('Sku')['Total_Payment'].transform('min')

# max
df['Max_tgt_sku']=df.groupby('Sku')['Total_Payment'].transform('max')

# std
df['Std_tgt_sku']=df.groupby('Sku')['Total_Payment'].transform('std')
```

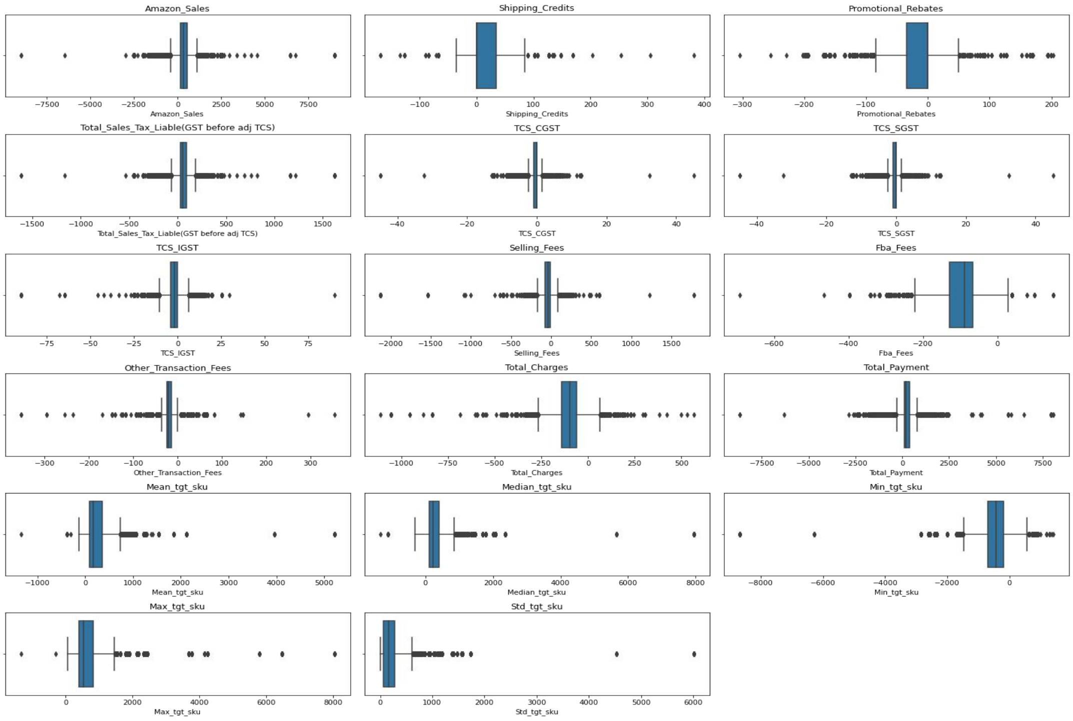
Heatmap after creating variables:



Inferences:

- Mean_tgt_sku and Median_tgt_sku is having positive correlation with target variable.

Outlier Detection:



Inferences:

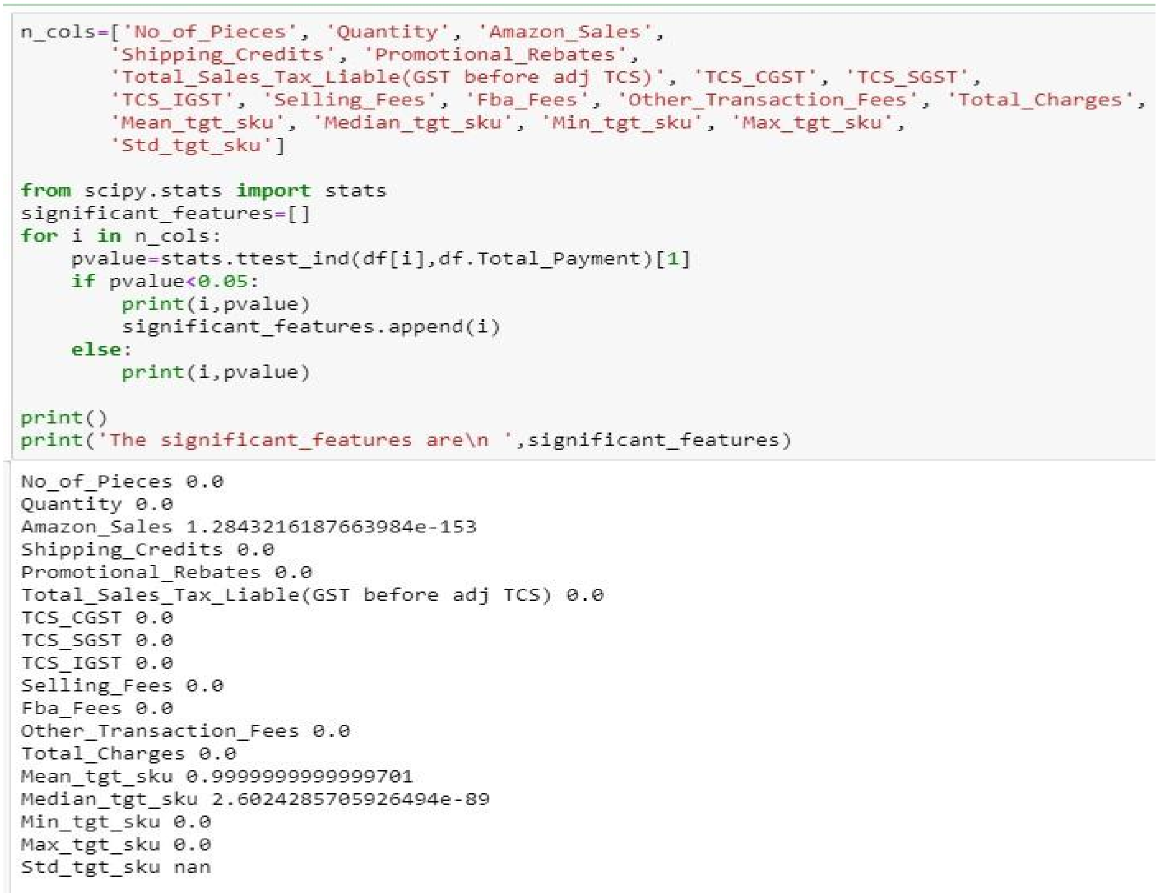
- From these above plots, we can visualize the outliers in the dataset. Since this is a sales dataset, we are not removing any outliers for further analysis.

STATISTICAL TESTS:

1. T - Test:

Hypothesis testing between Total_Payment feature and other numerical features

- H0: Both groups have equal mean indicating that they are insignificant.
- H1: Both groups do not have equal mean indicating that they are significant.



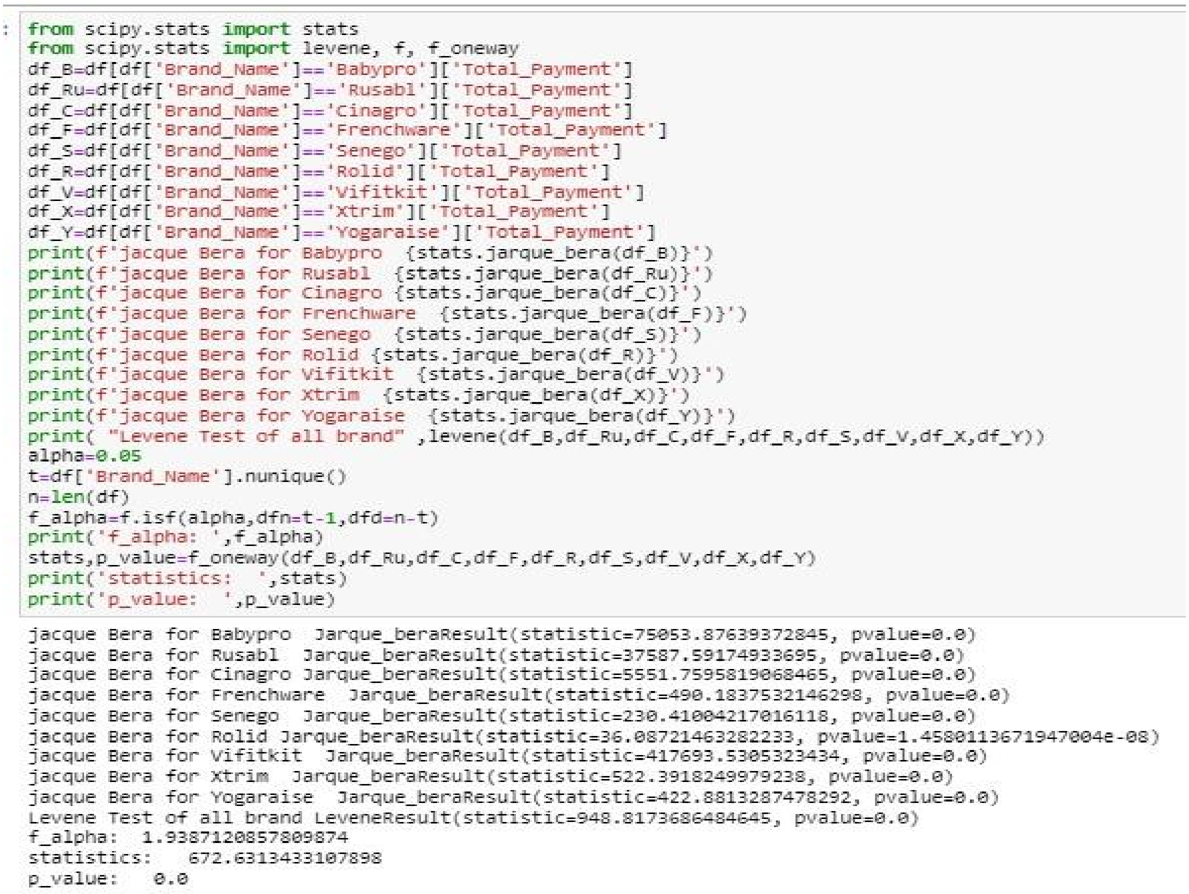
```
n_cols=['No_of_Pieces', 'Quantity', 'Amazon_Sales',
       'Shipping_Credits', 'Promotional_Rebates',
       'Total_Sales_Tax_Liable(GST before adj TCS)', 'TCS_CGST', 'TCS_SGST',
       'TCS_IGST', 'Selling_Fees', 'Fba_Fees', 'Other_Transaction_Fees', 'Total_Charges',
       'Mean_tgt_sku', 'Median_tgt_sku', 'Min_tgt_sku', 'Max_tgt_sku',
       'Std_tgt_sku']

from scipy.stats import stats
significant_features=[]
for i in n_cols:
    pvalue=stats.ttest_ind(df[i],df.Total_Payment)[1]
    if pvalue<0.05:
        print(i,pvalue)
        significant_features.append(i)
    else:
        print(i,pvalue)

print()
print('The significant_features are\n ',significant_features)
```

```
No_of_Pieces 0.0
Quantity 0.0
Amazon_Sales 1.2843216187663984e-153
Shipping_Credits 0.0
Promotional_Rebates 0.0
Total_Sales_Tax_Liable(GST before adj TCS) 0.0
TCS_CGST 0.0
TCS_SGST 0.0
TCS_IGST 0.0
Selling_Fees 0.0
Fba_Fees 0.0
Other_Transaction_Fees 0.0
Total_Charges 0.0
Mean_tgt_sku 0.9999999999999701
Median_tgt_sku 2.6024285705926494e-89
Min_tgt_sku 0.0
Max_tgt_sku 0.0
Std_tgt_sku nan
```

- The significant features are No of Pieces, Quantity, Amazon Sales, Shipping Credits, Promotional Rebates, Total Sales Tax Liable (GST before adj TCS), TCS-CGST, TCS-SGST, TCS-IGST, Selling Fees, Fba Fees, Other Transaction Fees, Total Charges, Median_tgt_sku, Min_tgt_sku, Max_tgt_sku.

2. Anova test of Brand Name and total Payment:

- Ho: there is no effect of Brand Name on total fees
- Ha: there is effect of Brand Name on total fees



```
from scipy.stats import stats
from scipy.stats import levene, f, f_oneway
df_B=df[df['Brand_Name']=='Babypro']['Total_Payment']
df_Ru=df[df['Brand_Name']=='Rusabl']['Total_Payment']
df_C=df[df['Brand_Name']=='Cinagro']['Total_Payment']
df_F=df[df['Brand_Name']=='Frenchware']['Total_Payment']
df_S=df[df['Brand_Name']=='Senego']['Total_Payment']
df_R=df[df['Brand_Name']=='Rolid']['Total_Payment']
df_V=df[df['Brand_Name']=='Vifitkit']['Total_Payment']
df_X=df[df['Brand_Name']=='Xtrim']['Total_Payment']
df_Y=df[df['Brand_Name']=='Yogaraise']['Total_Payment']
print(f'jacque Bera for Babypro  {stats.jarque_bera(df_B)}')
print(f'jacque Bera for Rusabl  {stats.jarque_bera(df_Ru)}')
print(f'jacque Bera for Cinagro {stats.jarque_bera(df_C)}')
print(f'jacque Bera for Frenchware  {stats.jarque_bera(df_F)}')
print(f'jacque Bera for Senego  {stats.jarque_bera(df_S)}')
print(f'jacque Bera for Rolid {stats.jarque_bera(df_R)}')
print(f'jacque Bera for Vifitkit  {stats.jarque_bera(df_V)}')
print(f'jacque Bera for Xtrim  {stats.jarque_bera(df_X)}')
print(f'jacque Bera for Yogaraise  {stats.jarque_bera(df_Y)}')
print( "Levene Test of all brand" ,levene(df_B,df_Ru,df_C,df_F,df_R,df_S,df_V,df_X,df_Y))
alpha=0.05
t=df['Brand_Name'].nunique()
n=len(df)
f_alpha=f.isf(alpha,dfn=t-1,dfd=n-t)
print('f_alpha: ',f_alpha)
stats,p_value=f_oneway(df_B,df_Ru,df_C,df_F,df_R,df_S,df_V,df_X,df_Y)
print('statistics:  ',stats)
print('p_value:  ',p_value)
```

```
jacque Bera for Babypro  Jarque_beraResult(statistic=75053.87639372845, pvalue=0.0)
jacque Bera for Rusabl  Jarque_beraResult(statistic=37587.59174933695, pvalue=0.0)
jacque Bera for Cinagro Jarque_beraResult(statistic=5551.7595819068465, pvalue=0.0)
jacque Bera for Frenchware  Jarque_beraResult(statistic=490.1837532146298, pvalue=0.0)
jacque Bera for Senego  Jarque_beraResult(statistic=230.41004217016118, pvalue=0.0)
jacque Bera for Rolid Jarque_beraResult(statistic=36.08721463282233, pvalue=1.4580113671947004e-08)
jacque Bera for Vifitkit  Jarque_beraResult(statistic=417693.5305323434, pvalue=0.0)
jacque Bera for Xtrim  Jarque_beraResult(statistic=522.3918249979238, pvalue=0.0)
jacque Bera for Yogaraise  Jarque_beraResult(statistic=422.8813287478292, pvalue=0.0)
Levene Test of all brand LeveneResult(statistic=948.8173686484645, pvalue=0.0)
f_alpha:  1.9387120857809874
statistics:   672.6313433107898
p_value:   0.0
```

Inferences:

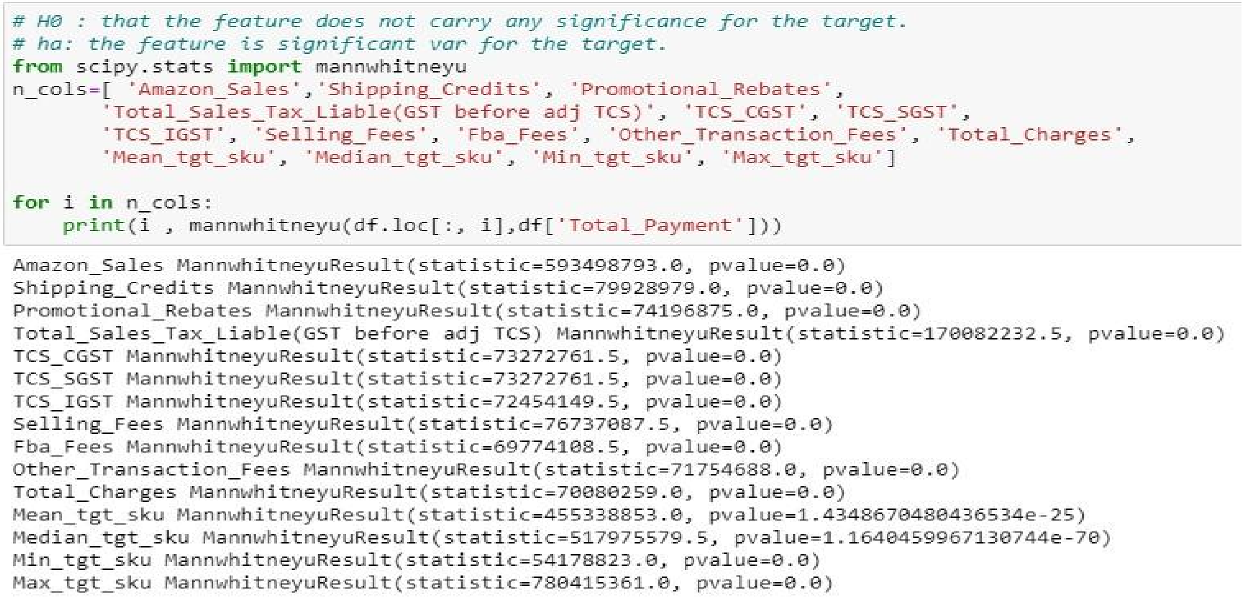
- By Jarque-Bera test, we can see that data is not normally distributed.
- By Levene test, we see that variance is equal.
- Since P-value < 0.05 and f_alpha < statistics, we reject the null hypothesis, that is, hence there is affect of Brand Name on total fees.

NON-PARAMETRIC TEST:

1. Mannwhitneyu:

Since the data is not normally distributed, we are proceeding with non-parametric test.

- Ho: The feature does not carry any significance for the target.
- Ha: The feature is significant variable for the target.



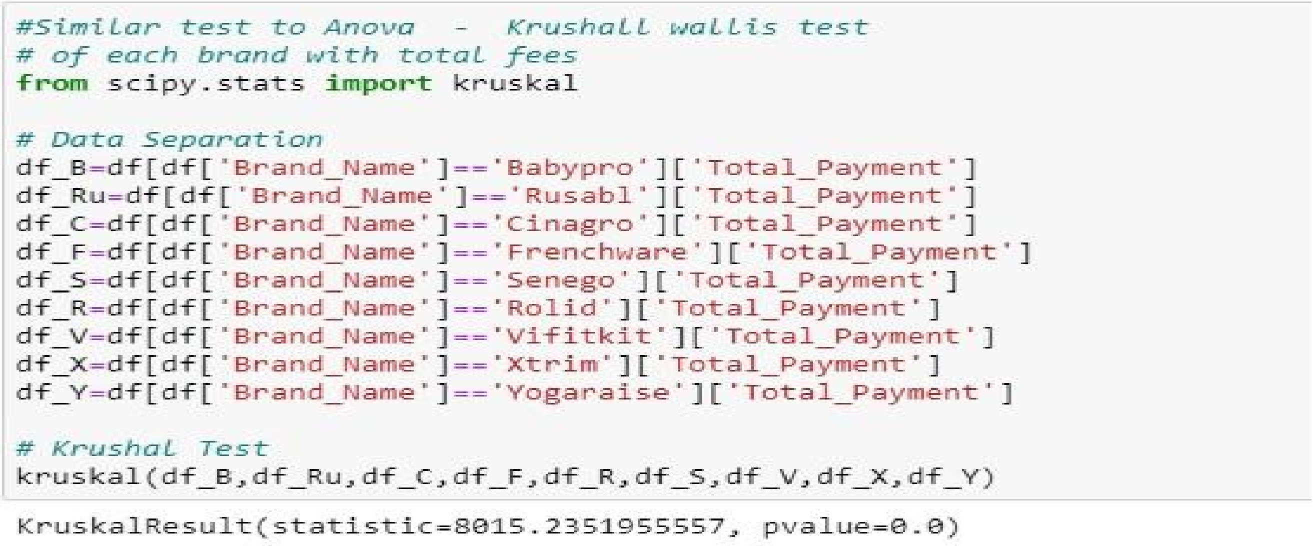
```
# H0 : that the feature does not carry any significance for the target.
# ha: the feature is significant var for the target.
from scipy.stats import mannwhitneyu
n_cols=[ 'Amazon_Sales','Shipping_Credits', 'Promotional_Rebates',
       'Total_Sales_Tax_Liable(GST before adj TCS)', 'TCS_CGST', 'TCS_SGST',
       'TCS_IGST', 'Selling_Fees', 'Fba_Fees', 'Other_Transaction_Fees', 'Total_Charges',
       'Mean_tgt_sku', 'Median_tgt_sku', 'Min_tgt_sku', 'Max_tgt_sku']

for i in n_cols:
    print(i , mannwhitneyu(df.loc[:, i],df['Total_Payment']))
```

```
Amazon_Sales MannwhitneyuResult(statistic=593498793.0, pvalue=0.0)
Shipping_Credits MannwhitneyuResult(statistic=79928979.0, pvalue=0.0)
Promotional_Rebates MannwhitneyuResult(statistic=74196875.0, pvalue=0.0)
Total_Sales_Tax_Liable(GST before adj TCS) MannwhitneyuResult(statistic=170082232.5, pvalue=0.0)
TCS_CGST MannwhitneyuResult(statistic=73272761.5, pvalue=0.0)
TCS_SGST MannwhitneyuResult(statistic=73272761.5, pvalue=0.0)
TCS_IGST MannwhitneyuResult(statistic=72454149.5, pvalue=0.0)
Selling_Fees MannwhitneyuResult(statistic=76737087.5, pvalue=0.0)
Fba_Fees MannwhitneyuResult(statistic=69774108.5, pvalue=0.0)
Other_Transaction_Fees MannwhitneyuResult(statistic=71754688.0, pvalue=0.0)
Total_Charges MannwhitneyuResult(statistic=70080259.0, pvalue=0.0)
Mean_tgt_sku MannwhitneyuResult(statistic=455338853.0, pvalue=1.4348670480436534e-25)
Median_tgt_sku MannwhitneyuResult(statistic=517975579.5, pvalue=1.1640459967130744e-70)
Min_tgt_sku MannwhitneyuResult(statistic=54178823.0, pvalue=0.0)
Max_tgt_sku MannwhitneyuResult(statistic=780415361.0, pvalue=0.0)
```

Inferences:

- Since, the p-value < 0.05, for all numerical variables, we reject the numerical variables.
- Hence, it means these variables are significant for the target. So, no columns to be dropped before model building.

2. Kruskal Test:



```
#Similar test to Anova  -  Krushall wallis test
# of each brand with total fees
from scipy.stats import kruskal

# Data Separation
df_B=df[df['Brand_Name']=='Babypro']['Total_Payment']
df_Ru=df[df['Brand_Name']=='Rusabl']['Total_Payment']
df_C=df[df['Brand_Name']=='Cinagro']['Total_Payment']
df_F=df[df['Brand_Name']=='Frenchware']['Total_Payment']
df_S=df[df['Brand_Name']=='Senego']['Total_Payment']
df_R=df[df['Brand_Name']=='Rolid']['Total_Payment']
df_V=df[df['Brand_Name']=='Vifitkit']['Total_Payment']
df_X=df[df['Brand_Name']=='Xtrim']['Total_Payment']
df_Y=df[df['Brand_Name']=='Yogaraise']['Total_Payment']

# Krushal Test
kruskal(df_B,df_Ru,df_C,df_F,df_R,df_S,df_V,df_X,df_Y)
```

```
KruskalResult(statistic=8015.2351955557, pvalue=0.0)
```

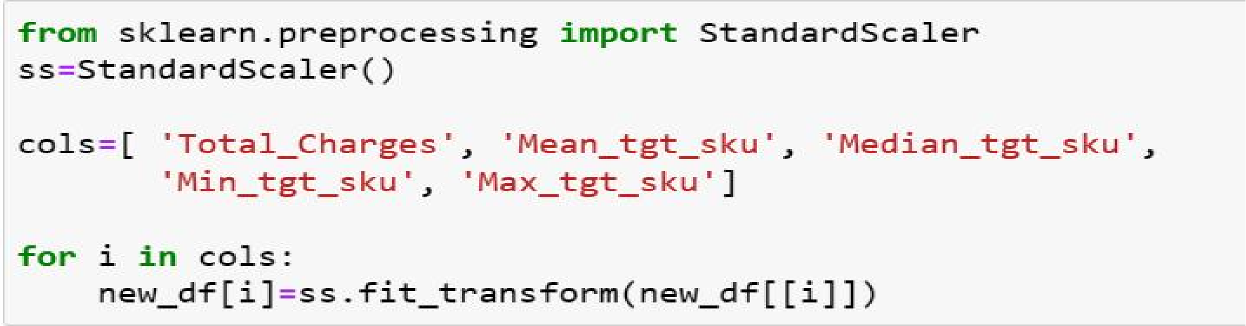
Inferences:

- Since, p-value < 0.05, both non-parametric test, that means all variables are significant.

Treated the redundant variables:

We dropped the redundant columns for further analysis and assigned to a new variable. After dropping the redundant columns, the dataset has 30937 rows and 15 columns.

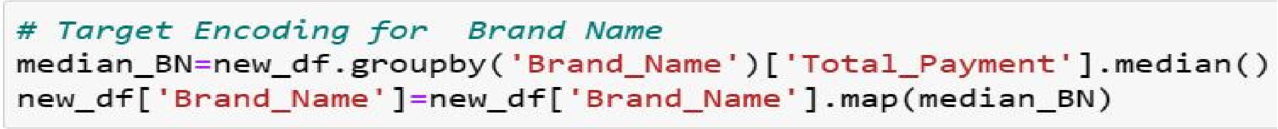
Scaling the desired attributes:



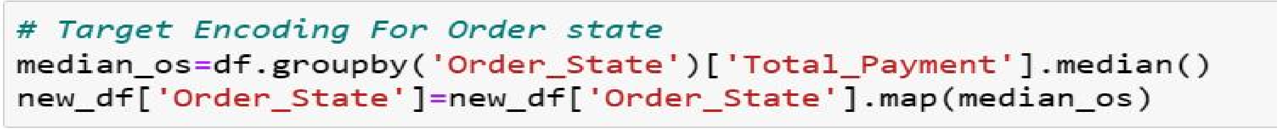
```
from sklearn.preprocessing import StandardScaler
ss=StandardScaler()

cols=[ 'Total_Charges', 'Mean_tgt_sku', 'Median_tgt_sku',
       'Min_tgt_sku', 'Max_tgt_sku']

for i in cols:
    new_df[i]=ss.fit_transform(new_df[[i]])
```

Using Standard Scaler we scaled Total Charges, Mean_tgt_sku, Median_tgt_sku, Min_tgt_sku Max_tgt_sku.

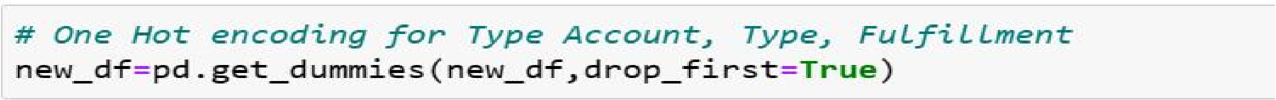
Encoding:



```
# Target Encoding for  Brand Name
median_BN=new_df.groupby('Brand_Name')['Total_Payment'].median()
new_df['Brand_Name']=new_df['Brand_Name'].map(median_BN)
```



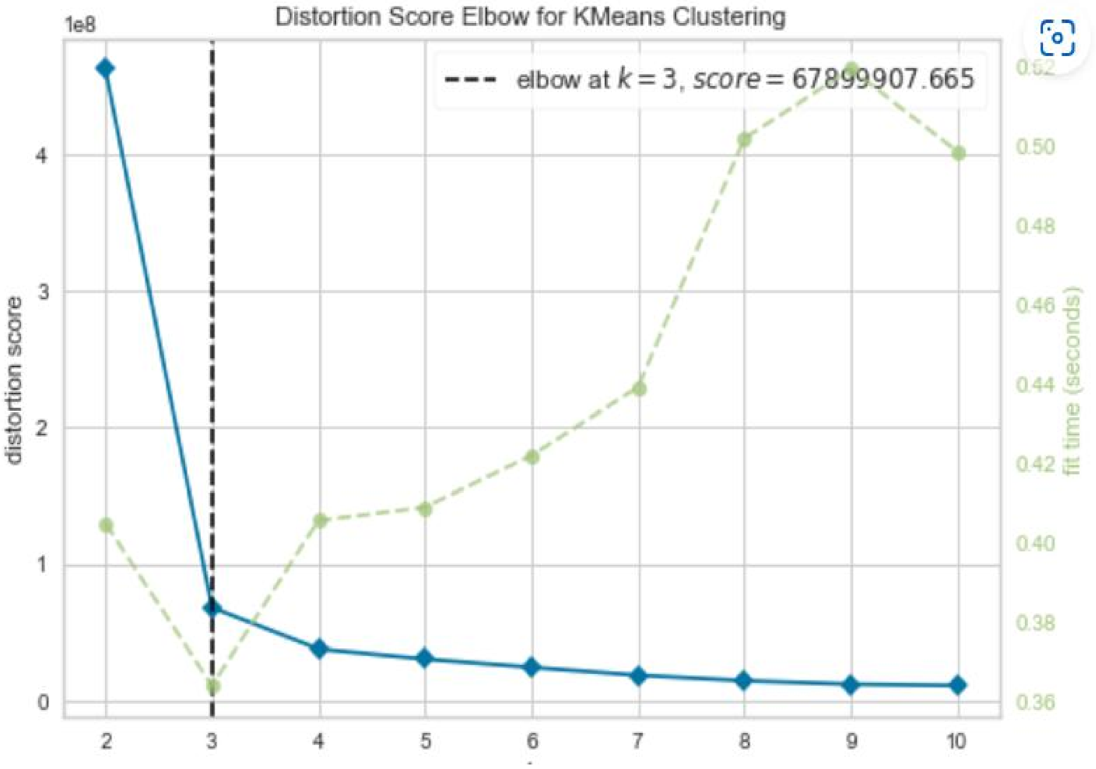
```
# Target Encoding For Order state
median_os=df.groupby('Order_State')['Total_Payment'].median()
new_df['Order_State']=new_df['Order_State'].map(median_os)
```



```
# One Hot encoding for Type Account, Type, Fulfillment
new_df=pd.get_dummies(new_df,drop_first=True)
```

- We had done Target encoding for Brand Name with Median values of Total Payment with respect to each Brand Name.
- We had done Target encoding for Order State with Median values of Total Payment with respect to each Order State.
- We had done One Hot encoding for Type Account, Type, Fulfilment.

SEGMENTATION:

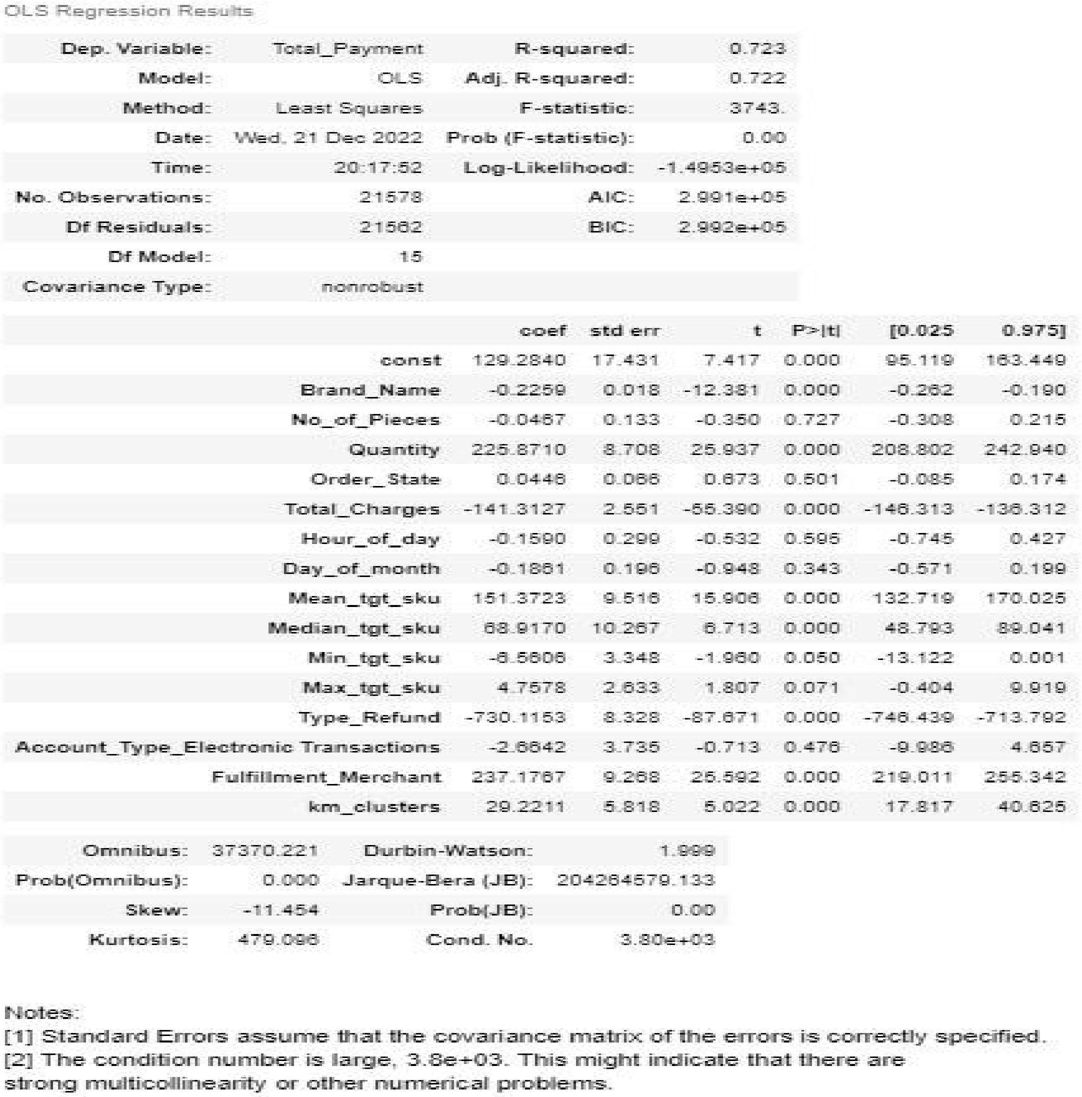


Inference:

- Using K-Elbow Visualizer for the data of Brand Name, Order State, Hour of day, Day of month, we found that specific number of cluster for these columns are Three.
- Then we done the segmentation for these data with 3 clusters as Labels

MODEL BUILDING:

Linear Regression – Statistical Model:



```
OLS Regression Results
   Dep. Variable:    Total_Payment          R-squared:       0.723
           Model:              OLS      Adj. R-squared:       0.722
          Method:    Least Squares         F-statistic:       3743.
            Date: Wed, 21 Dec 2022   Prob (F-statistic):       0.00
            Time:         20:17:52      Log-Likelihood: -1.4953e+05
No. Observations:            21578                 AIC:   2.991e+05
    Df Residuals:            21562                 BIC:   2.992e+05
        Df Model:               15
 Covariance Type:        nonrobust

                                           coef  std err        t   P>|t|    [0.025    0.975]
                              const    129.2840   17.431    7.417   0.000    95.119   163.449
                         Brand_Name     -0.2259    0.018  -12.381   0.000    -0.262    -0.190
                       No_of_Pieces     -0.0467    0.133   -0.350   0.727    -0.308     0.215
                           Quantity    225.8710    8.708   25.937   0.000   208.802   242.940
                        Order_State      0.0446    0.066    0.673   0.501    -0.085     0.174
                      Total_Charges   -141.3127    2.551  -55.390   0.000  -146.313  -136.312
                        Hour_of_day     -0.1590    0.299   -0.532   0.595    -0.745     0.427
                       Day_of_month     -0.1861    0.196   -0.948   0.343    -0.571     0.199
                       Mean_tgt_sku    151.3723    9.516   15.906   0.000   132.719   170.025
                     Median_tgt_sku     68.9170   10.267    6.713   0.000    48.793    89.041
                        Min_tgt_sku     -6.5606    3.348   -1.960   0.050   -13.122     0.001
                        Max_tgt_sku      4.7578    2.633    1.807   0.071    -0.404     9.919
                        Type_Refund   -730.1153    8.328  -87.671   0.000  -746.439  -713.792
Account_Type_Electronic Transactions    -2.6642    3.735   -0.713   0.476    -9.986     4.657
               Fulfillment_Merchant    237.1767    9.268   25.592   0.000   219.011   255.342
                        km_clusters     29.2211    5.818    5.022   0.000    17.817    40.625

      Omnibus: 37370.221     Durbin-Watson:         1.999
Prob(Omnibus):     0.000  Jarque-Bera (JB): 204264579.133
         Skew:   -11.454          Prob(JB):          0.00
     Kurtosis:   479.096          Cond. No.      3.80e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.8e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Inference:

- We found that R-squared value for Linear Regression as base model is 0.723 and Adj R-Squared value is 0.722.
- The Significance variable is Brand Name, Quantity, Total charges, Mean_tgt_sku, Median_tgt_sku, Type Refund, Fulfillment Merchant, km_clusters.

Assumption test:

1. Linearity – Rainbow Test:

Hypothesis:

Ho: data has linear relation with target variable.

Ha: data do not have linear relation with target variable.

- Since the p-value < 0.05 data doesn't have linear relation with target variable.

2. Normality test - Jacque bera test:

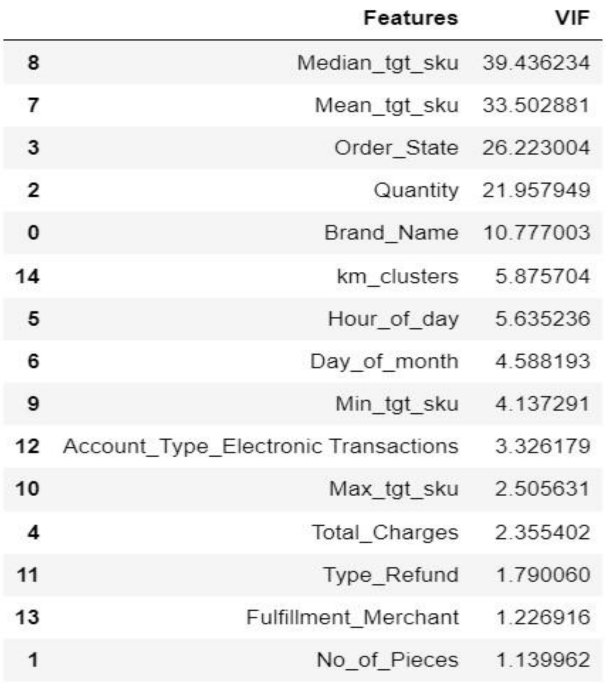
Hypothesis:

Ho: Data is normal.

Ha: Data is not normal.

- Since P_value < 0.05, the data is not normally distributed.

3. Multicollinearity Test – VIF:



	Features	VIF
8	Median_tgt_sku	39.436234
7	Mean_tgt_sku	33.502881
3	Order_State	26.223004
2	Quantity	21.957949
0	Brand_Name	10.777003
14	km_clusters	5.875704
5	Hour_of_day	5.635236
6	Day_of_month	4.588193
9	Min_tgt_sku	4.137291
12	Account_Type_Electronic Transactions	3.326179
10	Max_tgt_sku	2.505631
4	Total_Charges	2.355402
11	Type_Refund	1.790060
13	Fulfillment_Merchant	1.226916
1	No_of_Pieces	1.139962

- Median_tgt_sku and Mean_tgt_sku are having high multicollinearity.

4. Heteroscedasticity – Breusch Pagan Test:

Hypothesis:

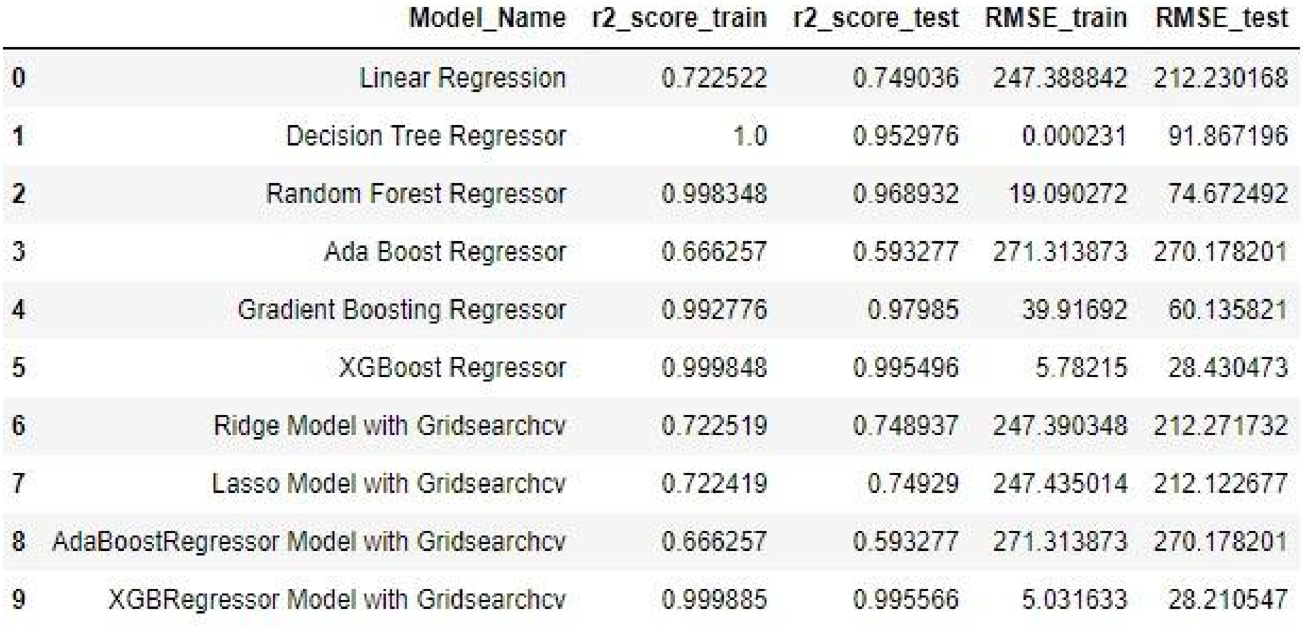
- Ho: That there is equal variance present in the data.
- Ha: There is unequal variance.

Inference: Since the p-value < 0.05, we reject the null hypothesis. That is the data has Heteroscedasticity.

5. Autocorrelation of Errors – Durbin Watson Test:

Since, Durbin-Watson value is 1.99, It has a positive autocorrelation of errors.

MODEL BUILDING RESULTS:



	Model_Name	r2_score_train	r2_score_test	RMSE_train	RMSE_test
0	Linear Regression	0.722522	0.749036	247.388842	212.230168
1	Decision Tree Regressor	1.0	0.952976	0.000231	91.867196
2	Random Forest Regressor	0.998348	0.968932	19.090272	74.672492
3	Ada Boost Regressor	0.666257	0.593277	271.313873	270.178201
4	Gradient Boosting Regressor	0.992776	0.97985	39.91692	60.135821
5	XGBoost Regressor	0.999848	0.995496	5.78215	28.430473
6	Ridge Model with Gridsearchcv	0.722519	0.748937	247.390348	212.271732
7	Lasso Model with Gridsearchcv	0.722419	0.74929	247.435014	212.122677
8	AdaBoostRegressor Model with Gridsearchcv	0.666257	0.593277	271.313873	270.178201
9	XGBRegressor Model with Gridsearchcv	0.999885	0.995566	5.031633	28.210547

Inferences:

- **For Linear regression model we found that R2 score for train is 0.72 and r2 score for test data is 0.74, RMSE for train in 247.38 and test is 212.23**
- **For Ada boost regressor model we found that R2 score for train is 0.66 and r2 score for test data is 0.59, RMSE for train in 271.31 and test is 270.17**
- **For Gradiant Boost regressor model we found that R2 score for train is 0.99 and r2 score for test data is 0.97, RMSE for train in 39.91 and test is 60.13**

- **For XG Boost regressor model we found that R2 score for train is 0.99 and r2 score for test data is 0.99, RMSE for train in 5.78 and test is 28.43**
- **For Ridge model we found that R2 score for train is 0.72 and r2 score for test data is 0.74, RMSE for train in 247.39 and test is 212.27**

Ada Boost Regressor and Gradiant Boost Regressor are good models compared to other models.

Business Inference:

- We found that Cinagro have highest sales in Harayana compared to other brands and Senego and Xtrim have the negative sales in Megalaya because they have more return orders so company has to focus on those type of orders.

- Rusabl is having more customer service issue and loss inbound orders compared to other brands, so they need to work on customer service.

- Amazon sales of these brands are very less in Andaman n Nicobar , Ladakh and Damen and Diu, so they need to focus on marketing activities.

- We found that Cinagro have highest sales than other brands ,so they do have more FBA Inventory Reimbursement (Customer Return). So they can open a Storage facility and they can delivery products at least possible time and there will be less product inbounds and if still damage happens, they can make replacement of that product instead of giving a refund to the customer.