

E-retail factors for customer activation and retention

(A case study from Indian e-commerce customers)

Prepared by:
Abhishek Ranjan

Contents

- 1. Problem statement and understanding**
- 2. EDA Steps, visualizations and assumptions used to complete the project**
- 3. Analysis, Observation & Conclusion.**

Problem statement : Factors for growth of Online Store

- ❖ Nowadays, e-Commerce or Online store is one of the growing business.
- ❖ Many businessmen and entrepreneurs : focused on rapid expansion of Online store.
- ❖ From customer point of view, Online store will not only reduce people's time but also give a large option at one place to chose the correct product.
- ❖ However, success of Online store or e-Store is most importantly dependent on ***Customer satisfaction***. It is one of the most important factors that guarantees its success.
- ❖ Customer satisfaction is a key stimulant of :
 - i. purchase,
 - ii. repurchase intentions, and
 - iii. customer loyalty.

Problem statement : Factors for growth of Online Store

- ❖ Five major factors that contributed to the success of an e-commerce store have been identified as:
 - a. service quality,
 - b. system quality,
 - c. information quality,
 - d. Trust, and
 - e. net benefit.



Need to identify Factors for Online Store

- ❖ The research investigated the factors that influence the online customers repeat purchase intention.
- ❖ The combination of both utilitarian value and hedonistic values : needed to affect the repeat purchase intention (loyalty) positively.

Method for study of Factors for Online Store growth

- ❖ A comprehensive review of the literature, theories and models have been carried out to propose the models for customer activation and customer retention.
- ❖ The data is collected from the Indian online shoppers.
- ❖ Results will indicate the e-retail success factors, which : very much critical for customer satisfaction.

Steps and assumptions for data study

Importing Libraries

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

Importing the DATASET

In [2]:

```
retention=pd.read_excel("Customer_retention_dataset.xlsx")
```

Retrieving Datasheet

In [4]: #To print all columns

```
pd.set_option('display.max_columns',None)
pd.set_option('display.max_rows',None)
```

In [5]: retention.head()

out[5]:

1Gender of respondent	2 How old are you?	3 Which city do you shop online from?	4 What is the Pin Code of where you shop online from?	5 Since How Long You are Shopping Online ?	6 How many times you have made an online purchase in the past 1 year?	7 How do you access the internet while shopping on-line?	8 Which device do you use to access the online shopping?	9 What is the screen size of your mobile device?	10 What is the operating system (OS) of your device?	11 What browser do you run on your device to access the website?	12 Which channel did you follow to arrive at your favorite online store for the first time?	13 After first visit, how do you reach the online retail store?	14 How much time do you explore the e-retail store before making a purchase decision?	15 V
0	Male 31-40 years	Delhi 110009	Above 4 years	31-40 times	Dial-up	Desktop	Others	Window/windows Mobile	Google chrome	Search Engine	Search Engine	6-10 mins	E- (Free)	
1	Female 21-30 years	Delhi 110030	Above 4 years	41 times and above	Wi-Fi	Smartphone	4.7 inches	iOS/Mac	Google chrome	Search Engine	Via application	more than 15 mins	Cred	
2	Female 21-30 years	Greater Noida 201308	3-4 years	41 times and above	Mobile Internet	Smartphone	5.5 inches	Android	Google chrome	Search Engine	Via application	11-15 mins	E- (Free)	
3	Male 21-30 years	Karnal 132001	3-4 years	Less than 10 times	Mobile Internet	Smartphone	5.5 inches	iOS/Mac	Safari	Search Engine	Search Engine	6-10 mins	Cred	
4	Female 21-30 years	Bangalore 530068	2-3 years	11-20 times	Wi-Fi	Smartphone	4.7 inches	iOS/Mac	Safari	Content Marketing	Via application	more than 15 mins	Cred	

EDA Steps and assumptions for data study

EDA

Checking Total Numbers of Rows and Column

```
In [6]: retention.shape  
out[6]: (269, 71)
```

Checking All Column Name

EDA Steps and assumptions for data study

Before further Analysis, will Rename Column Names to understand it better as column names are too large.

EDA Steps and assumptions for data study

```
Out[8]: Index(['Gender', 'Age', 'City_Of_Shopping', 'Pin_Code',
   'Online_Shopping_Since', 'Purchase_Frequency', 'Internet_Access_Mode',
   'Device_Used', 'Mobile_Screen_Size', 'Device_Operating_System',
   'Browser_Used', 'Channel_Followed', 'Login_Mode',
   'Exploring_Time_Spent', 'Payment_Mode',
   'Shopping_Cart_Abandon_Frequency', 'Bag_Abandon_Reason',
   'Content_Readability', 'Similar_Product_Info', 'Seller_Product_Info',
   'Product_Info_Clarity', 'Website_Navigation_Ease',
   'Loading_Processing_Speed', 'User_Friendly_Interface',
   'Conveninet_Payment_Mode', 'Timely_Fulfilment_Trust',
   'Customer_Empathy', 'Customer_Privacy_Guarantee',
   'Several_Channels_Responses', 'Discount_Benefit',
   'Online_Shopping_Enjoyment', 'Online_Shopping_Convenience_Flexibility',
   'Return_Replace_Policy', 'Loyalty_Program_Access',
   'Quality_Information_Satisfaction', 'Quality_Satisfaction',
   'Net_Benefit_Satisfaction', 'Users_Trust', 'Product_Categories',
   'Relevant_Product_Information', 'Monetary_Savings',
   'Patronizing_Convenience', 'Adventure_Sense', 'Enhances_Social_Status',
   'Shopping_Gratification', 'Role_Fulfilment', 'Worth_of_Money',
   'Shopped_From_Retailer', 'Ease_Website_Application',
   'Visually_Appealing_layout', 'Product_Variety',
   'Complete_Product_Information', 'Fast>Loading_Speed_Web_App',
   'Reliable_Website_Application', 'Quick_Purchase', 'Payment_Options',
   'Fast_Delivery', 'Customer_Info_Privacy',
   'Customer_Financial_Info_Security', 'Perceived_Trustworthiness',
   'Multi_Channel_Assistance', 'Loading_Logging_Time',
   'Loading_Graphics_Photo_Display_Time', 'Late_Price_Declare',
   'Page>Loading_Time', 'Limited_Payment_Mode', 'Late_Delivery',
   'Design_Change_of_Web_App', 'Page_Disruption', 'Website_Efficiency',
   'Recommend_to_Friend'],
  dtype='object')
```

EDA Steps and assumptions for data study

Checking Data Type of All Data

```
In [10]: retention.dtypes
```

```
Out[10]:
```

Gender	object
Age	object
City_Of_Shopping	object
Pin_Code	int64
Online_Shopping_Since	object
Purchase_Frequency	object
Internet_Access_Mode	object
Device_Used	object
Mobile_Screen_Size	object
Device_Operating_System	object
Browser_Used	object
Channel_Followed	object
Login_Mode	object
Exploring_Time_Spent	object
Payment_Mode	object
Shopping_Cart_Abandon_Frequency	object
Bag_Abandon_Reason	object
Content_Readability	object
Similar_Product_Info	object
Seller_Product_Info	object
Product_Info_Clarity	object
Website_Navigation_Ease	object
Loading_Processing_Speed	object
User_Friendly_Interface	object
Conveninet_Payment_Mode	object
Timely_Fulfilment_Trust	object
Customer_Empathy	object
Customer_Privacy_Guarantee	object
Several_Channels_Responses	object
Discount_Benefit	object
Online_Shopping_Enjoyment	object
Online_Shopping_Convenience_Flexibility	object
Return_Replace_Policy	object
Loyalty_Program_Access	object
Quality_Information_Satisfaction	object
Quality_Satisfaction	object
Net_Benefit_Satisfaction	object
Users_Trust	object
Product_Categories	object



Checking for Null Values

```
In [11]: retention.isnull().sum()
```

```
Out[11]:
```

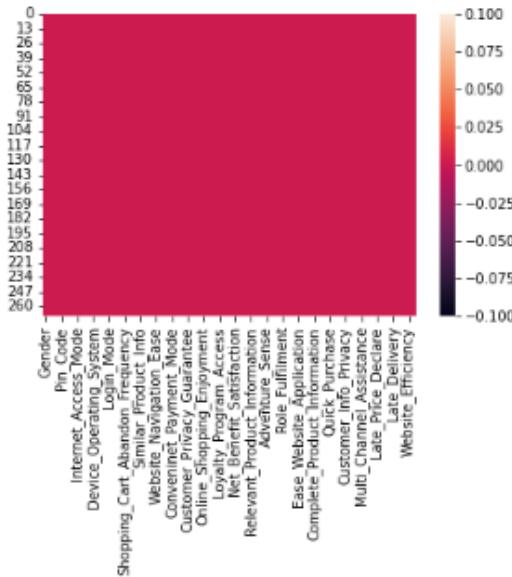
Gender	0
Age	0
City_Of_Shopping	0
Pin_Code	0
Online_Shopping_Since	0
Purchase_Frequency	0
Internet_Access_Mode	0
Device_Used	0
Mobile_Screen_Size	0
Device_Operating_System	0
Browser_Used	0
Channel_Followed	0
Login_Mode	0
Exploring_Time_Spent	0
Payment_Mode	0
Shopping_Cart_Abandon_Frequency	0
Bag_Abandon_Reason	0
Content_Readability	0
Similar_Product_Info	0
Seller_Product_Info	0
Product_Info_Clarity	0
Website_Navigation_Ease	0
Loading_Processing_Speed	0
User_Friendly_Interface	0
Conveninet_Payment_Mode	0
Timely_Fulfilment_Trust	0
Customer_Empathy	0
Customer_Privacy_Guarantee	0
Several_Channels_Responses	0
Discount_Benefit	0
Online_Shopping_Enjoyment	0
Online_Shopping_Convenience_Flexibility	0
Return_Replace_Policy	0
Loyalty_Program_Access	0
Quality_Information_Satisfaction	0
Quality_Satisfaction	0
Net_Benefit_Satisfaction	0
Users_Trust	0
Product_Categories	0
Relevant_Product_Information	0

EDA Steps and assumptions for data study

Checking for Null Values through heatmap also

```
In [12]: sns.heatmap(retention.isnull())
```

```
Out[12]: <AxesSubplot:>
```



Here also we can see there is no Null Value present in dataset.

Information about Data (Memory Used and Data Types)

```
In [13]: retention.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 269 entries, 0 to 268
```

```
Data columns (total 71 columns):
```

#	Column	Non-Null Count	Dtype
0	Gender	269	non-null
1	Pin_Code	269	object
2	Internet_Access_Mode	269	non-null
3	Device_Operating_System	269	non-null
4	Login_Mode	269	non-null
5	Customer_Privacy_Guarantee	269	non-null
6	Online_Shopping_Enjoyment	269	non-null
7	Loyalty_Program_Access	269	non-null
8	Net_Benefit_Satisfaction	269	non-null
9	Relevant_Product_Information	269	non-null
10	Adventure_Sense_Role_Fulfillment	269	non-null
11	Website_Navigation_Ease	269	non-null
12	Customer_Privacy_Guarantee	269	non-null
13	Complete_Product_Information	269	non-null
14	Quick_Purchase	269	non-null
15	Customer_Info_Privacy	269	non-null
16	Multi_Channel_Assistance	269	non-null
17	Late_Price_Disclaimer	269	non-null
18	Late_Delivery	269	non-null
19	Website_Efficiency	269	non-null
20	Relevant_Product_Information	269	non-null
21	Adventure_Sense_Role_Fulfillment	269	non-null
22	Website_Navigation_Ease	269	non-null
23	Customer_Privacy_Guarantee	269	non-null
24	Complete_Product_Information	269	non-null
25	Quick_Purchase	269	non-null
26	Customer_Info_Privacy	269	non-null
27	Multi_Channel_Assistance	269	non-null
28	Late_Price_Disclaimer	269	non-null
29	Late_Delivery	269	non-null
30	Website_Efficiency	269	non-null
31	Relevant_Product_Information	269	non-null
32	Adventure_Sense_Role_Fulfillment	269	non-null

EDA Steps and assumptions for data study

Checking if 0 values present in dataset or not

```
In [14]: (retention=='?').sum()
```

```
Out[14]: Gender          0  
Age             0  
City_Of_Shopping  0  
Pin_Code        0  
Online_Shopping_Since 0  
Purchase_Frequency 0  
Internet_Access_Mode 0  
Device_Used      0  
Mobile_Screen_Size 0  
Device_Operating_System 0  
Browser_Used     0  
Channel_Followed 0  
Login_Mode       0  
Exploring_Time_Spent 0  
Payment_Mode     0  
Shopping_Cart_Abandon_Frequency 0  
Bag_Abandon_Reason 0  
Content_Readability 0  
Similar_Product_Info 0  
Seller_Product_Info 0  
Product_Info_Clarity 0  
Website_Navigation_Ease 0  
Loading_Processing_Speed 0  
User_Friendly_Interface 0  
Conveninet_Payment_Mode 0  
Timely_Fulfilment_Trust 0  
Customer_Empathy    0  
Customer_Privacy_Guarantee 0  
Several_Channels_Responses 0  
Discount_Benefit    0  
Online_Shopping_Enjoyment 0  
Online_Shopping_Convenience_Flexibility 0  
Return_Replace_Policy 0  
Loyalty_Program_Access 0  
Quality_Information_satisfaction 0  
Quality_Satisfaction 0  
Net_Benefit_Satisfaction 0  
Users_Trust        0  
Product_Categories 0
```

Checking total number of unique value in each column of Dataset

```
In [15]: retention.nunique()
```

```
Out[15]: Gender          2  
Age             5  
City_Of_Shopping  11  
Pin_Code        39  
Online_Shopping_Since 5  
Purchase_Frequency 6  
Internet_Access_Mode 4  
Device_Used      4  
Mobile_Screen_Size 4  
Device_Operating_System 3  
Browser_Used     4  
Channel_Followed 3  
Login_Mode       5  
Exploring_Time_Spent 5  
Payment_Mode     3  
Shopping_Cart_Abandon_Frequency 4  
Bag_Abandon_Reason 5  
Content_Readability 4  
Similar_Product_Info 4  
Seller_Product_Info 5  
Product_Info_Clarity 4  
Website_Navigation_Ease 4  
Loading_Processing_Speed 5  
User_Friendly_Interface 5  
Conveninet_Payment_Mode 3  
Timely_Fulfilment_Trust 4  
Customer_Empathy    4  
Customer_Privacy_Guarantee 3  
Several_Channels_Responses 4  
Discount_Benefit    5  
Online_Shopping_Enjoyment 5  
Online_Shopping_Convenience_Flexibility 4  
Return_Replace_Policy 3  
Loyalty_Program_Access 5  
Quality_Information_Satisfaction 3  
Net_Benefit_Satisfaction 1
```

EDA Steps and assumptions for data study

We can see there are repeated values in two columns("Purchase_Frequency", "Internet_Access_Mode"). So we will remove duplicates value from these columns.

```
In [17]: #Column name "Purchase_Frequency" have value "41 times and above" and "42 times and above". Both value means same "41 times and above"
retention['Purchase_Frequency'].replace("42 times and above","41 times and above",inplace=True)
retention['Purchase_Frequency'].value_counts()
```

```
Out[17]: Less than 10 times    114
31-40 times      63
41 times and above  53
11-20 times      29
21-30 times      10
Name: Purchase_Frequency, dtype: int64
```

```
In [18]: #Column name "Internet_Access_Mode" have "Mobile internet" value 2 times("I" of "Internet" is in capital Letter in one and in small letter in another)
retention["Internet_Access_Mode"].replace("Mobile internet","Mobile Internet",inplace=True)
retention['Internet_Access_Mode'].value_counts()
```

```
Out[18]: Mobile Internet    189
Wi-Fi                76
Dial-up               4
Name: Internet_Access_Mode, dtype: int64
```

EDA Steps and assumptions for data study

Descriptive Statistics

```
In [19]: # Description of Dataset : works only on continuous column  
retention.describe()
```

Out[19]:

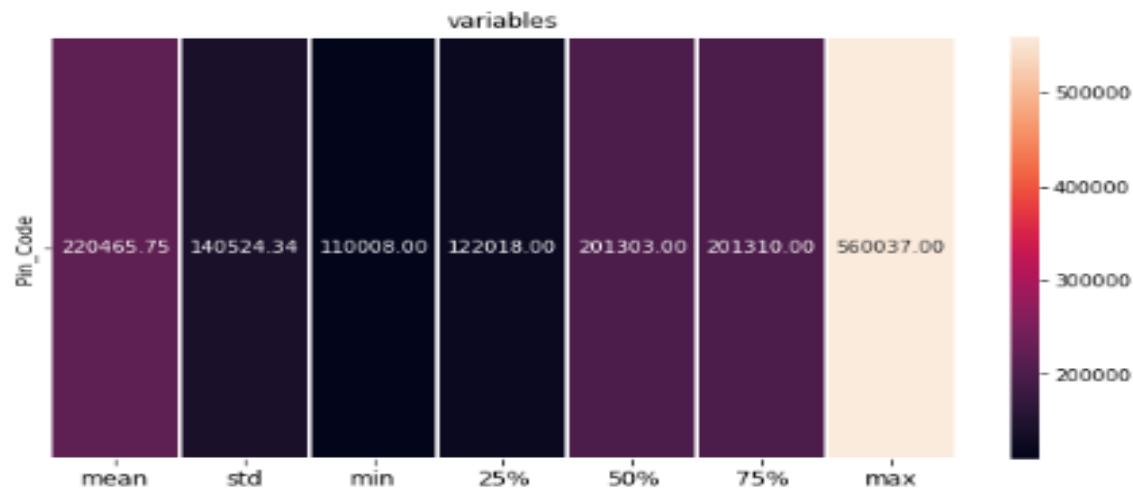
Pin_Code	
count	269.000000
mean	220485.747212
std	140524.341051
min	110008.000000
25%	122018.000000
50%	201303.000000
75%	201310.000000
max	560037.000000

- Only one column 'Pin_Code' is containing continuous data out of 71 columns.
- Total count is 269 and Total rows of DataSet is also 269. So, we can check that no null is present in this column.

EDA Steps and assumptions for data study

Checking Description through heatmap also.

```
In [20]: plt.figure(figsize=(10,5))
sns.heatmap(round(retention.describe()[1:]).transpose(),2), linewidth=2, annot=True, fmt='.2f')
plt.xticks(fontsize=18)
plt.xticks(fontsize=12)
plt.title('variables')
plt.show()
```



Outcome:

- Mean: 220465.747212
- std: 140524.341051
- min: 110008.000000
- max: 560037.000000

Data Visualization : Based on Gender

Data Visualization

Univariate Analysis

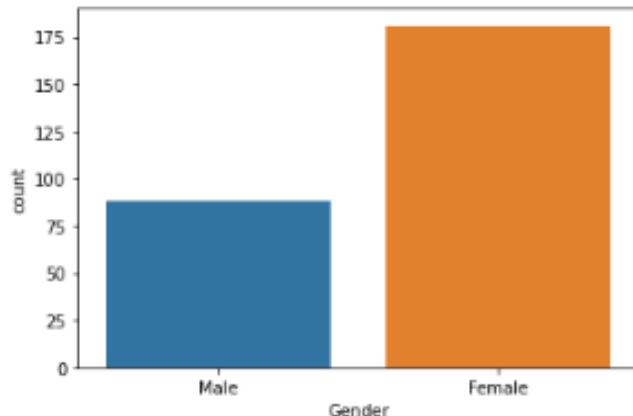
Using Countplot for categorical columns [¶](#)

```
In [22]: #Count Plot for "Gender" column  
print(retention["Gender"].value_counts())
```

```
sns.countplot("Gender",data=retention)
```

```
Female    181  
Male      88  
Name: Gender, dtype: int64
```

```
out[22]: <AxesSubplot:xlabel='Gender', ylabel='count'>
```



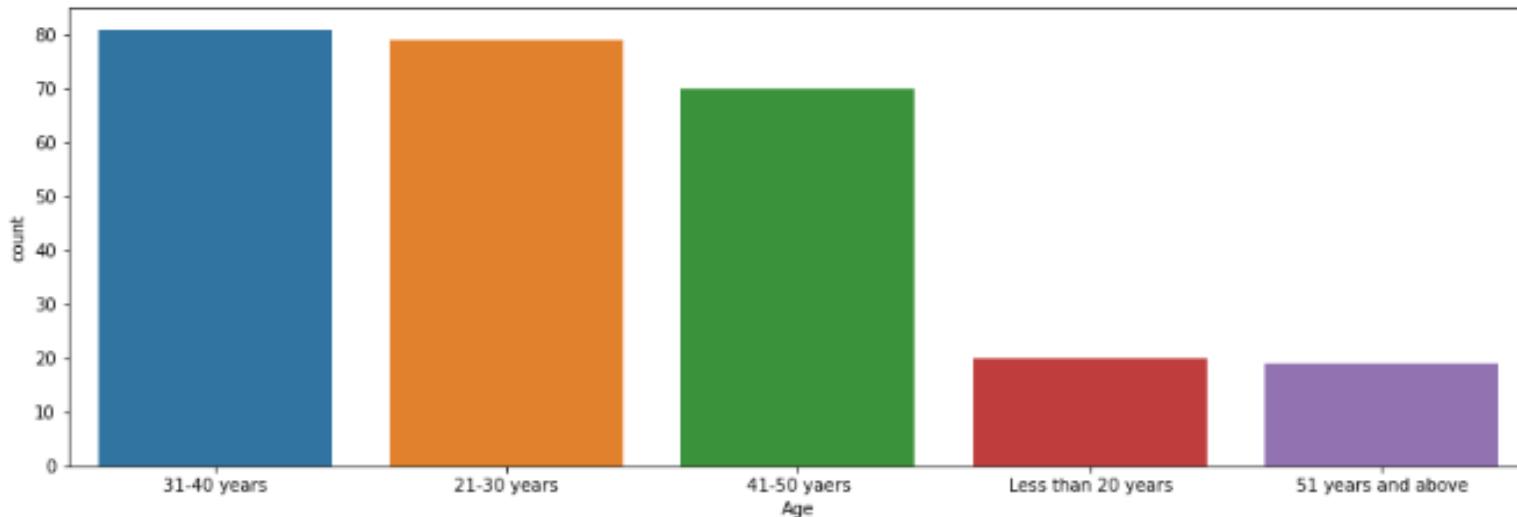
We observed Female customers are more compare to Male customers. Total Number of Female Customers are 181 and Male Customers are 88.

Data Visualization : Based on Age

```
In [23]: #Count Plot for "Age" column  
print(retention["Age"].value_counts())  
plt.figure(figsize=(15,5))  
sns.countplot("Age",data=retention)
```

```
31-40 years      81  
21-30 years      79  
41-50 yaers      70  
Less than 20 years    20  
51 years and above    19  
Name: Age, dtype: int64
```

```
Out[23]: <AxesSubplot:xlabel='Age', ylabel='count'>
```



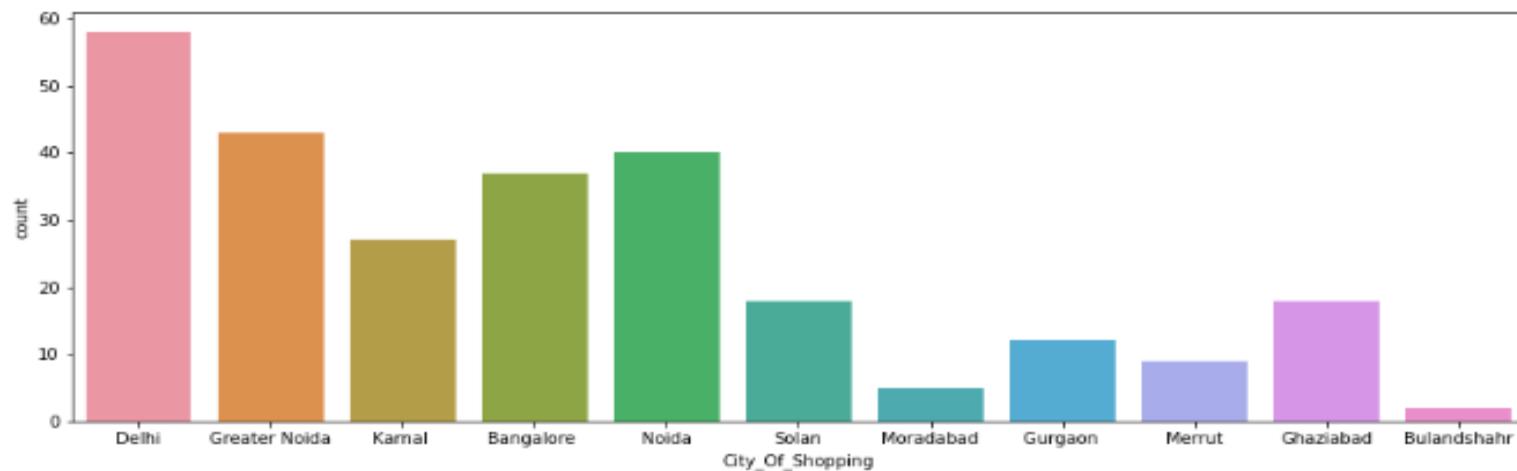
Customers having age more than 31-40 years are more (Total Number= 81) and having age more than 51 years and above are least (Total Number= 19) do Online Shopping.

Data Visualization : Based on City

```
In [24]: #Count Plot for "City_Of_Shopping" column  
print(retention["City_Of_Shopping"].value_counts())  
plt.figure(figsize=(15,5))  
sns.countplot("City_Of_Shopping",data=retention)
```

```
Delhi          58  
Greater Noida  43  
Noida          40  
Bangalore      37  
Karnal         27  
Solan          18  
Ghaziabad     18  
Gurgaon        12  
Merrut          9  
Moradabad      5  
Bulandshahr    2  
Name: City_Of_Shopping, dtype: int64
```

```
Out[24]: <AxesSubplot:xlabel='City_Of_Shopping', ylabel='count'>
```



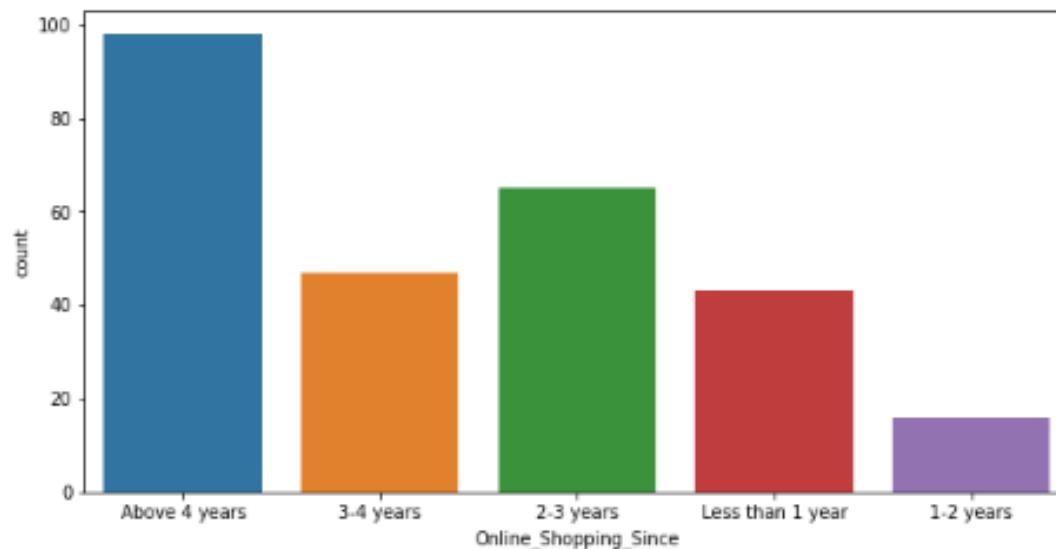
From "Delhi" Online Shopping was done most (Total Number= 58) and from "Bulandshahr" Online Shopping was done least (Total Number= 2).

Data Visualization : Based on Online Usage time

```
In [26]: #Count Plot for "Online_Shopping_Since" column  
print(retention["Online_Shopping_Since"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Online_Shopping_Since",data=retention)
```

```
Above 4 years      98  
2-3 years         65  
3-4 years         47  
Less than 1 year  43  
1-2 years          16  
Name: Online_Shopping_Since, dtype: int64
```

```
out[26]: <AxesSubplot:xlabel='Online_Shopping_Since', ylabel='count'>
```



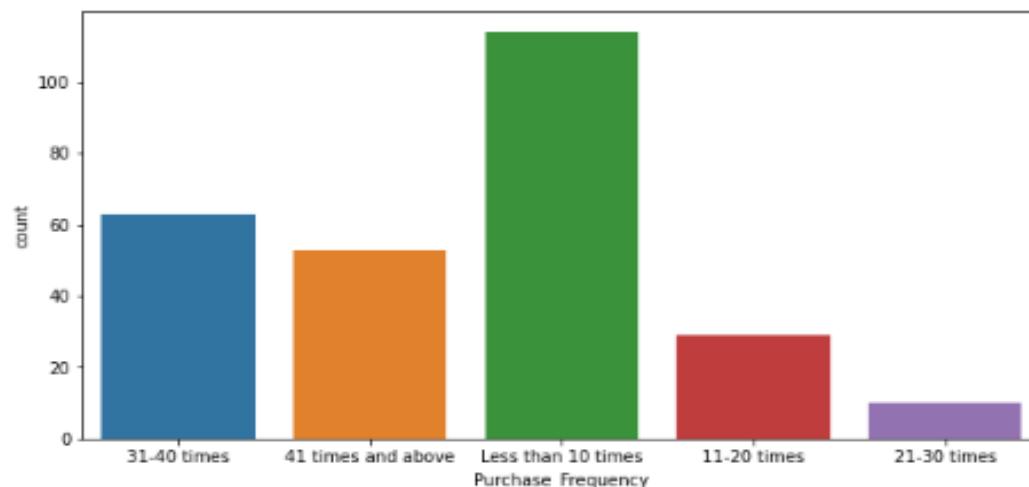
Doing Online Shopping from Above 4 years (Total No. 98) are most and doing Online Shopping from 1-2 years are least (Total No. 16)

Data Visualization : Based on Purchase Frequency

```
In [27]: #Count Plot for "Purchase_Frequency" column  
print(retention["Purchase_Frequency"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Purchase_Frequency",data=retention)
```

```
Less than 10 times      114  
31-40 times            63  
41 times and above     53  
11-20 times             29  
21-30 times              10  
Name: Purchase_Frequency, dtype: int64
```

```
Out[27]: <AxesSubplot:xlabel='Purchase_Frequency', ylabel='count'>
```



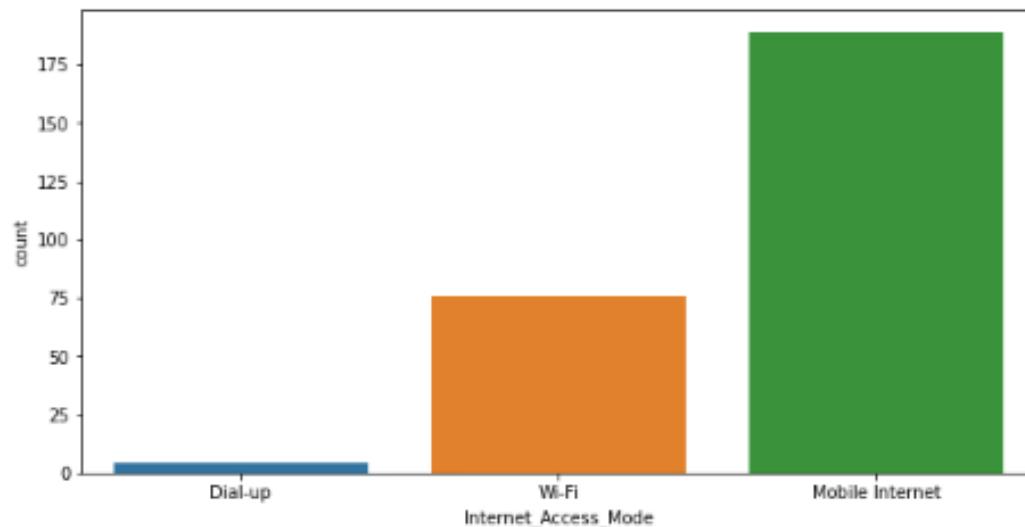
Online purchase in the past 1 year is Less than 10 times is most (Total No. 114) and Online purchase done 21-30 times is least (Total No. 10)

Data Visualization : Based on Internet Access Mode

```
In [28]: #Count Plot for "Internet_Access_Mode" column  
print(retention["Internet_Access_Mode"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Internet_Access_Mode",data=retention)
```

```
Mobile Internet    189  
Wi-Fi            76  
Dial-up           4  
Name: Internet_Access_Mode, dtype: int64
```

```
Out[28]: <AxesSubplot:xlabel='Internet_Access_Mode', ylabel='count'>
```



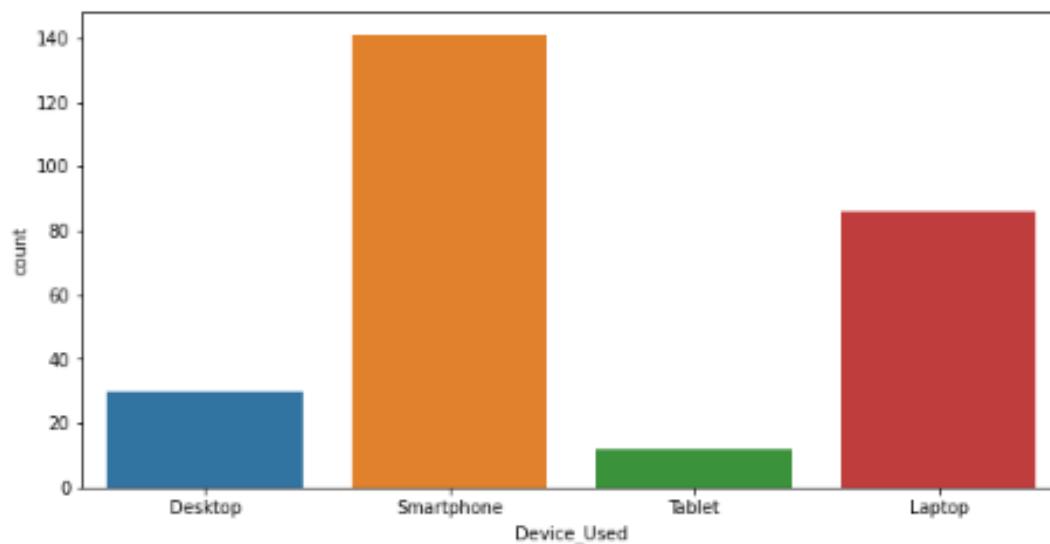
Shopping Online through Mobile Internet is more (Total No= 189) then done through Wi-Fi (Total No= 76) and done through Dial-up is least (Total No= 4)

Data Visualization : Based on Device Used

```
In [29]: #Count Plot for "Device_Used" column  
print(retention["Device_Used"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Device_Used",data=retention)
```

```
Smartphone    141  
Laptop        86  
Desktop       30  
Tablet        12  
Name: Device_Used, dtype: int64
```

```
Out[29]: <AxesSubplot:xlabel='Device_Used', ylabel='count'>
```



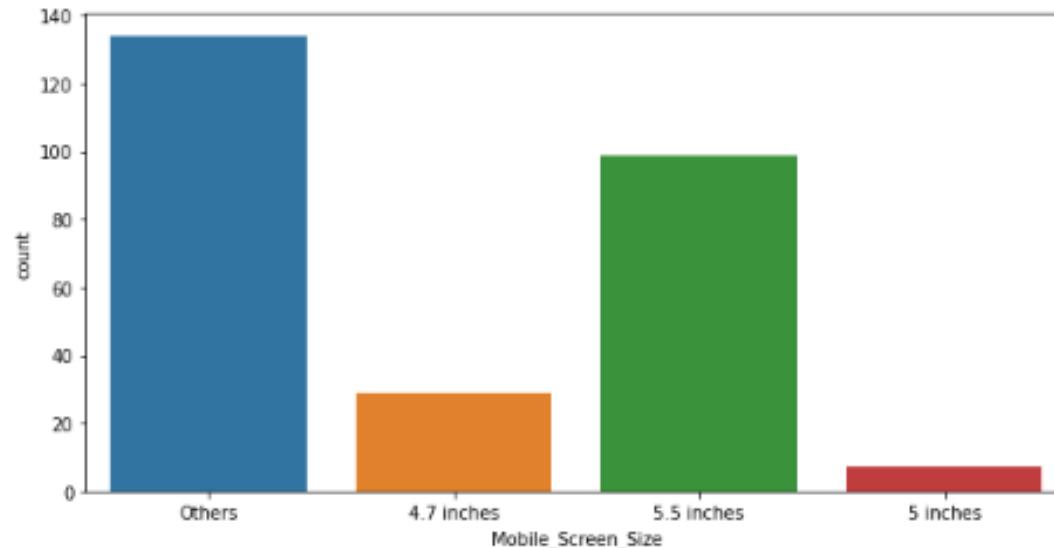
Device used to access the online shopping most is Smartphone (Total No= 141) and used least through Tablet (Total No= 12)

Data Visualization : Based on Mobile Screen Size

```
In [30]: #Count Plot for "Mobile_Screen_Size" column  
print(retention["Mobile_Screen_Size"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Mobile_Screen_Size",data=retention)
```

```
Others      134  
5.5 inches  99  
4.7 inches  29  
5 inches    7  
Name: Mobile_Screen_Size, dtype: int64
```

```
Out[30]: <AxesSubplot:xlabel='Mobile_Screen_Size', ylabel='count'>
```



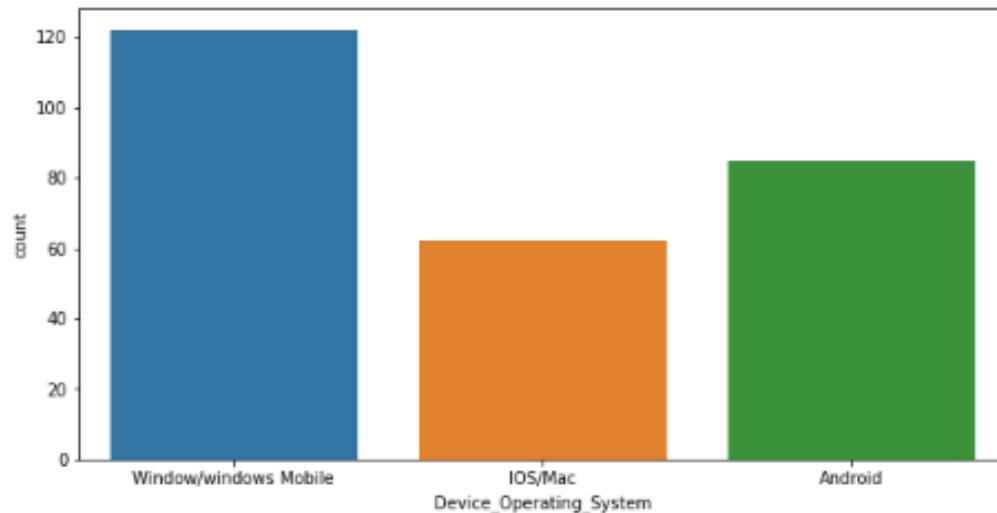
The screen size of mobile device "Others" is most (Total No= 134) and "5 inches" is least (Total No= 7)

Data Visualization : Based on Device OS

```
In [31]: #Count Plot for "Device_Operating_System" column  
print(retention["Device_Operating_System"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Device_Operating_System",data=retention)
```

```
Window/windows Mobile    122  
Android                  85  
IOS/Mac                  62  
Name: Device_Operating_System, dtype: int64
```

```
Out[31]: <AxesSubplot:xlabel='Device_Operating_System', ylabel='count'>
```



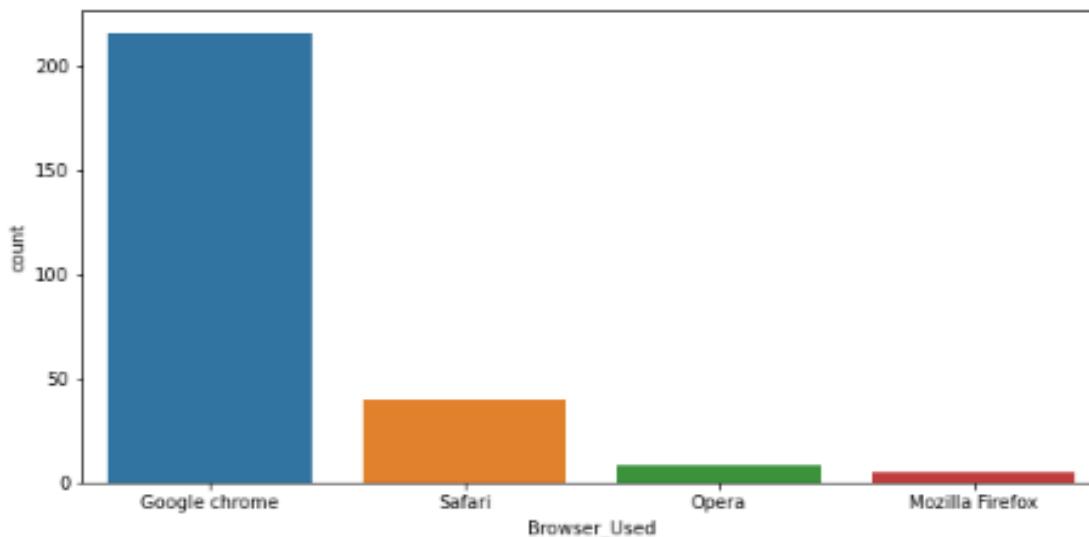
The operating system (OS) of device "Window/windows Mobile" is used most (Total No= 122) then Android (Total No= 85)and least is "IOS/Mac" is used least (Total No= 62)

Data Visualization : Based on Browser used

```
In [32]: #Count Plot for "Browser_Used" column  
print(retention["Browser_Used"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Browser_Used",data=retention)
```

```
Google chrome      216  
Safari             40  
Opera              8  
Mozilla Firefox    5  
Name: Browser_Used, dtype: int64
```

```
Out[32]: <AxesSubplot:xlabel='Browser_Used', ylabel='count'>
```



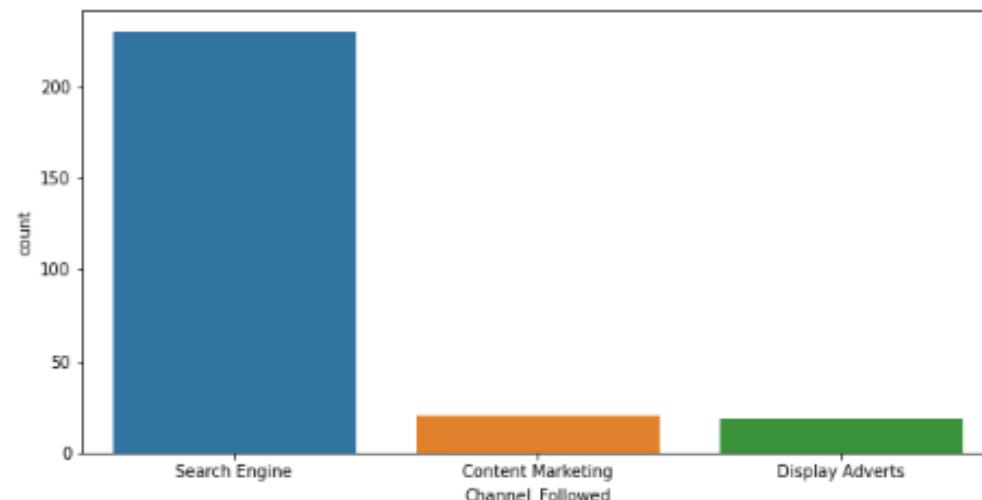
Browser used on device to access the website most is Google chrome (Total No= 216) and least is Mozilla Firefox (Total No= 5)

Data Visualization : Based on Channel Followed

```
In [33]: #Count Plot for "Channel_Followed" column  
print(retention["Channel_Followed"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Channel_Followed",data=retention)
```

```
Search Engine      230  
Content Marketing   20  
Display Adverts     19  
Name: Channel_Followed, dtype: int64
```

```
Out[33]: <AxesSubplot:xlabel='Channel_Followed', ylabel='count'>
```



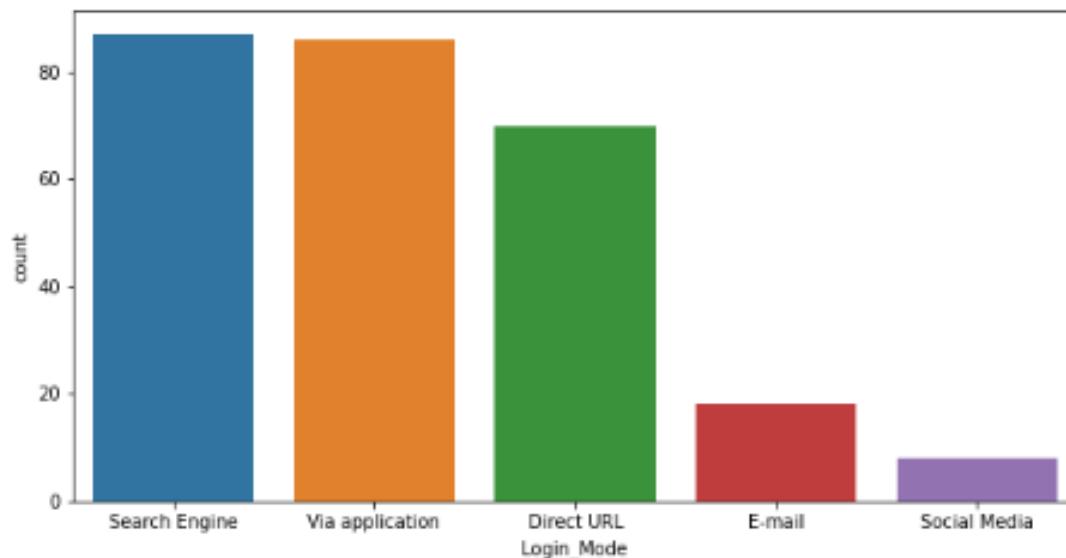
Channel followed to arrive at favorite online store for the first time is Search Engine (Total No= 230) then through Content Marketing (Total No= 20) and then through Display Adverts (Total No= 19)

Data Visualization : Based on Login mode

```
In [34]: #Count Plot for "Login_Mode" column  
print(retention["Login_Mode"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Login_Mode",data=retention)
```

```
Search Engine      87  
Via application   86  
Direct URL        70  
E-mail             18  
Social Media       8  
Name: Login_Mode, dtype: int64
```

```
out[34]: <AxesSubplot:xlabel='Login_Mode', ylabel='count'>
```



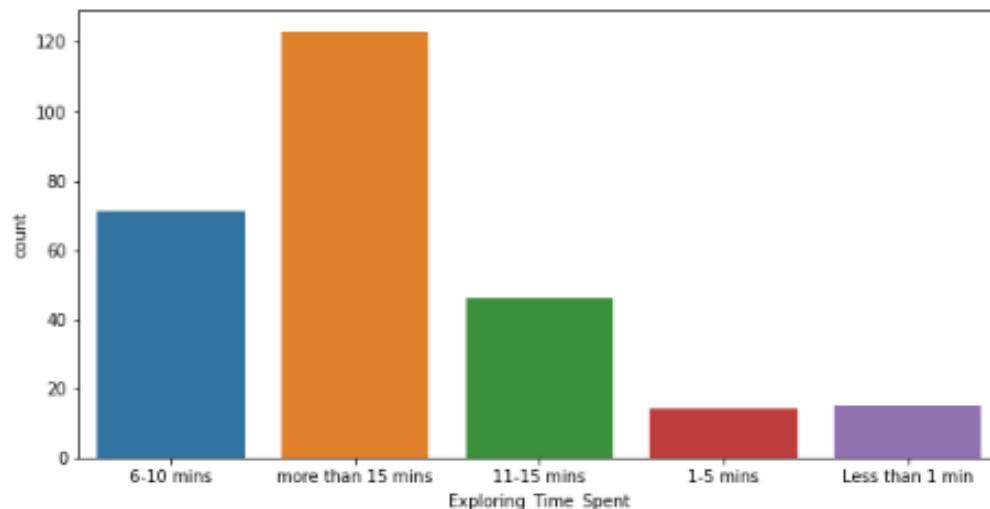
After first visit, Search Engine is used most (Total No= 87) to reach the online retail store and Social Media is used least (Total No= 8).

Data Visualization : Based on Exploring time

```
In [35]: #Count Plot for "Exploring_Time_Spent" column  
print(retention["Exploring_Time_Spent"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Exploring_Time_Spent",data=retention)
```

```
more than 15 mins    123  
6-10 mins          71  
11-15 mins         46  
Less than 1 min     15  
1-5 mins           14  
Name: Exploring_Time_Spent, dtype: int64
```

```
Out[35]: <AxesSubplot:xlabel='Exploring_Time_Spent', ylabel='count'>
```



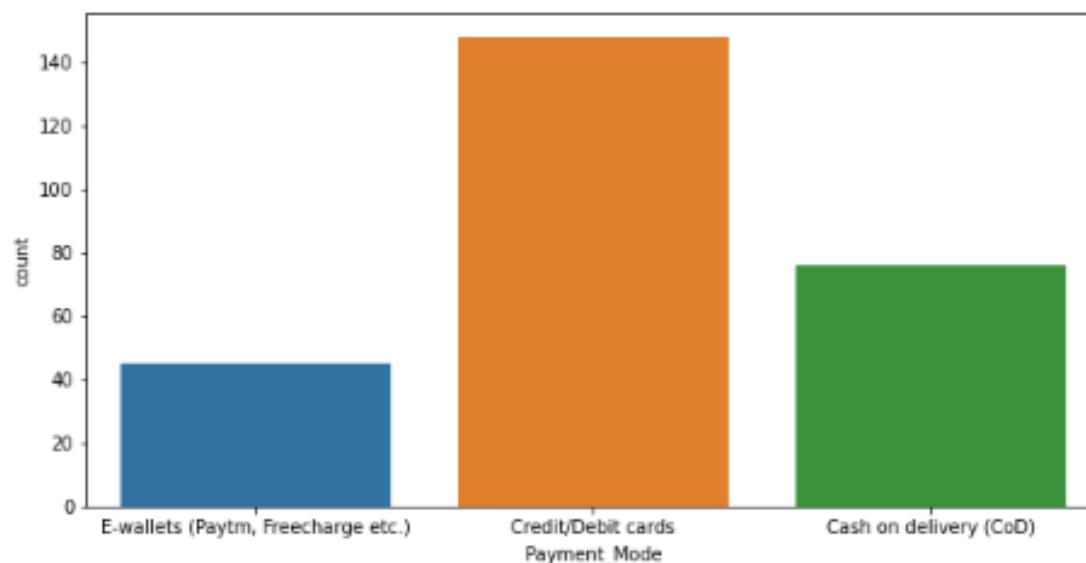
Most "more than 15 mins" (Total No= 123) time taken to explore the e-retail store before making a purchase decision and least "1-5 mins" (Total No= 14) to explore the e-retail store

Data Visualization : Based on Payment mode

```
In [36]: #Count Plot for "Payment_Mode" column  
print(retention["Payment_Mode"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Payment_Mode",data=retention)
```

```
Credit/Debit cards           148  
Cash on delivery (CoD)      76  
E-wallets (Paytm, Freecharge etc.) 45  
Name: Payment_Mode, dtype: int64
```

```
Out[36]: <AxesSubplot:xlabel='Payment_Mode', ylabel='count'>
```



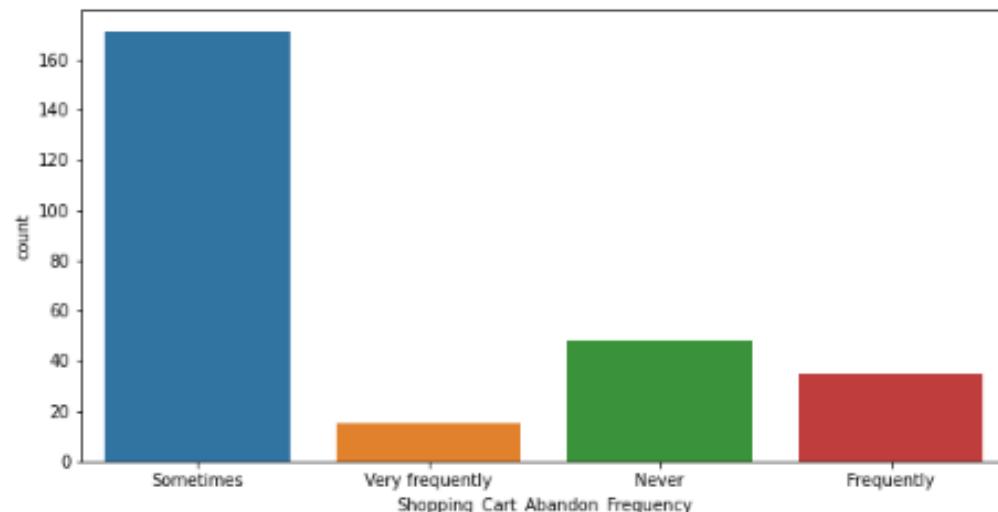
Credit/Debit cards (Total No= 148) is most Preferred Payment Option and E-wallets (Paytm, Freecharge etc.) (Total No= 45) is least

Data Visualization : Based on Shopping Cart Abandon frequency

```
In [37]: #Count Plot for "Shopping_Cart_Abandon_Frequency" column  
print(retention["Shopping_Cart_Abandon_Frequency"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Shopping_Cart_Abandon_Frequency",data=retention)
```

```
Sometimes      171  
Never          48  
Frequently     35  
Very frequently 15  
Name: Shopping_Cart_Abandon_Frequency, dtype: int64
```

```
Out[37]: <AxesSubplot:xlabel='Shopping_Cart_Abandon_Frequency', ylabel='count'>
```



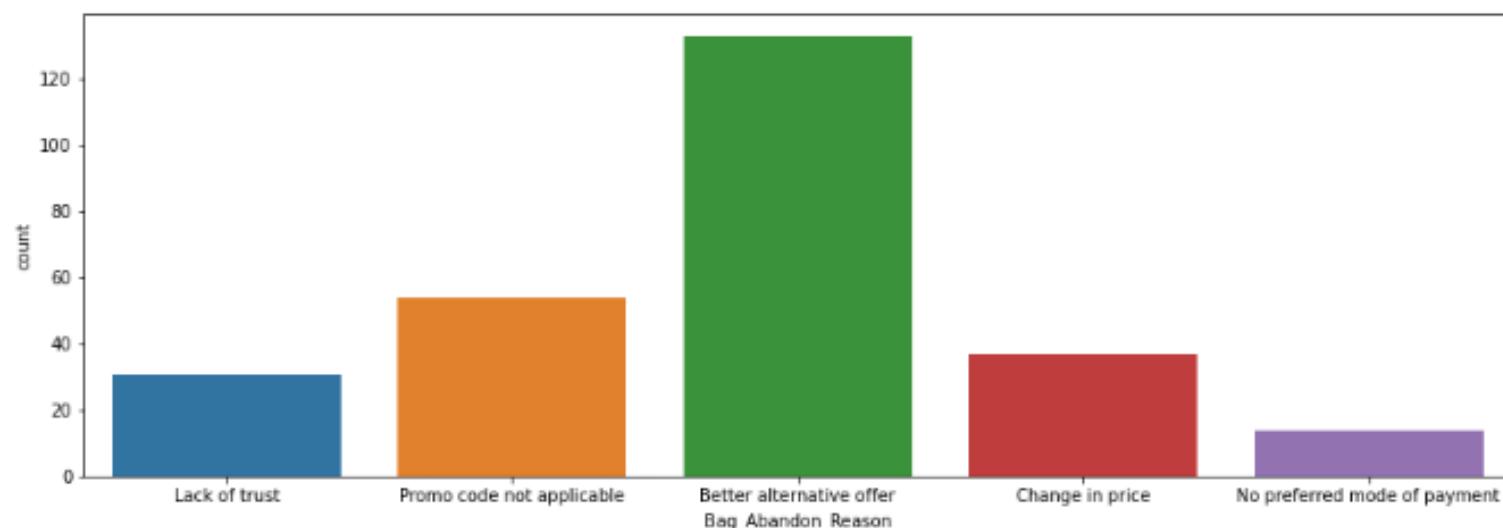
Abandoning (selecting an items and leaving without making payment) shopping cart frequently most is "Sometimes" (Total No= 171) and "Very frequently" (Total No= 15) is least.

Data Visualization : Based on Bag Abandon Reason

```
In [38]: #Count Plot for "Bag_Abandon_Reason" column  
print(retention["Bag_Abandon_Reason"].value_counts())  
plt.figure(figsize=(15,5))  
sns.countplot("Bag_Abandon_Reason",data=retention)
```

```
Better alternative offer      133  
Promo code not applicable   54  
Change in price              37  
Lack of trust                31  
No preferred mode of payment 14  
Name: Bag_Abandon_Reason, dtype: int64
```

```
out[38]: <AxesSubplot:xlabel='Bag_Abandon_Reason', ylabel='count'>
```



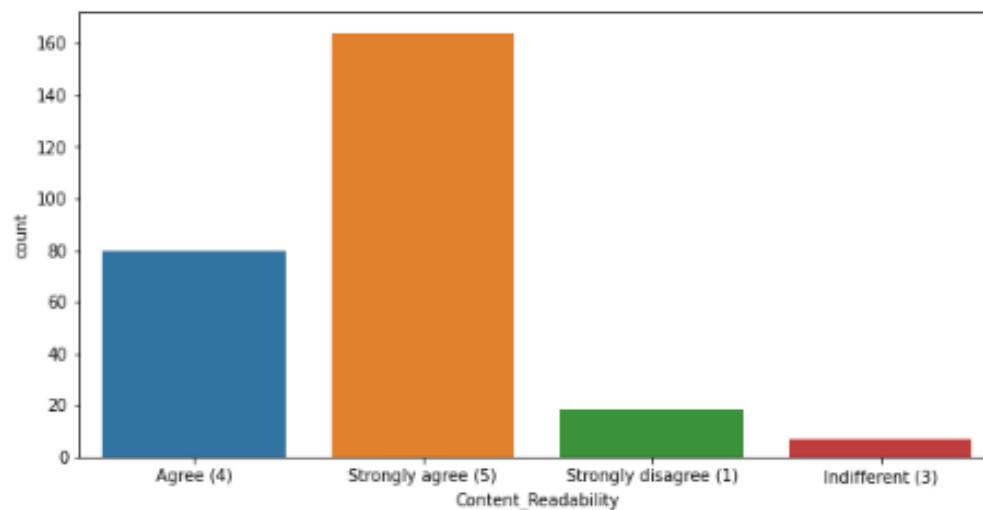
Reason to abandon the "Bag"/"Shopping Cart" most is Better alternative offer (Total No= 133) and least is No preferred mode of payment (Total No= 14)

Data Visualization : Based on Content Readability

```
In [39]: #Count Plot for "Content_Readability" column  
print(retention["Content_Readability"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Content_Readability",data=retention)
```

```
Strongly agree (5)      164  
Agree (4)              80  
Strongly disagree (1)  18  
Indifferent (3)        7  
Name: Content_Readability, dtype: int64
```

```
Out[39]: <AxesSubplot:xlabel='Content_Readability', ylabel='count'>
```



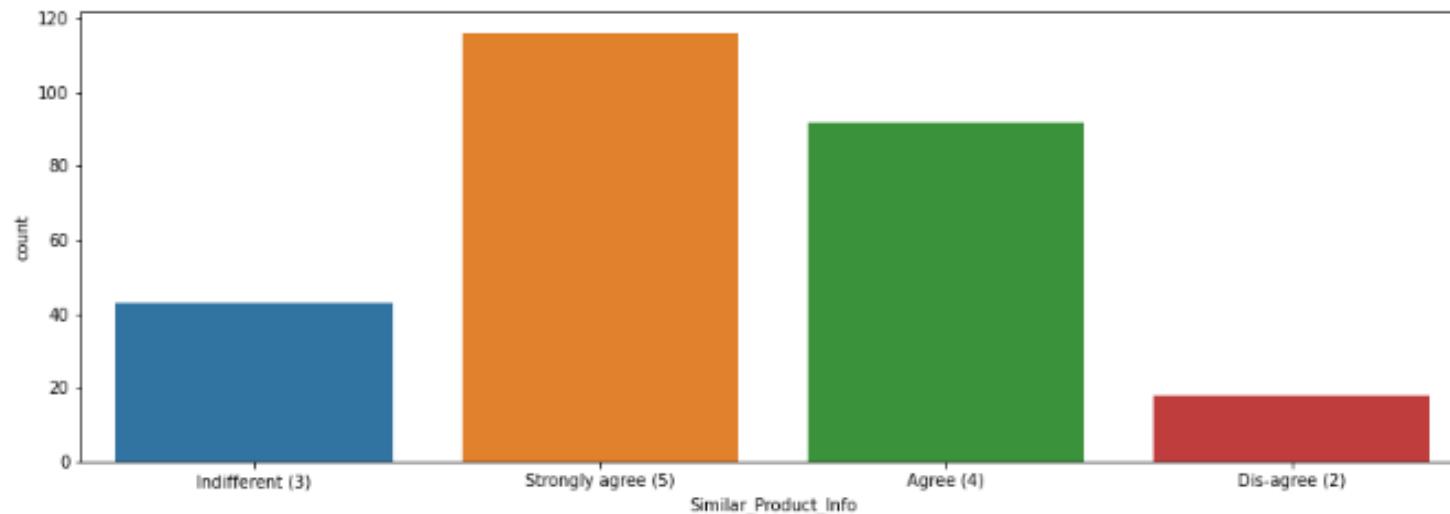
The content on the website must be easy to read and understand is Strongly agree most and got rating 5 for it from Total no of customers = 164 and Indifferent got rating 3 from only total no of 7 customers.

Data Visualization : Based on Similar product Info

```
In [40]: #Count Plot for "Similar_Product_Info" column  
print(retention["Similar_Product_Info"].value_counts())  
plt.figure(figsize=(15,5))  
sns.countplot("Similar_Product_Info",data=retention)
```

```
Strongly agree (5)    116  
Agree (4)           92  
Indifferent (3)     43  
Dis-agree (2)        18  
Name: Similar_Product_Info, dtype: int64
```

```
Out[40]: <AxesSubplot:xlabel='Similar_Product_Info', ylabel='count'>
```



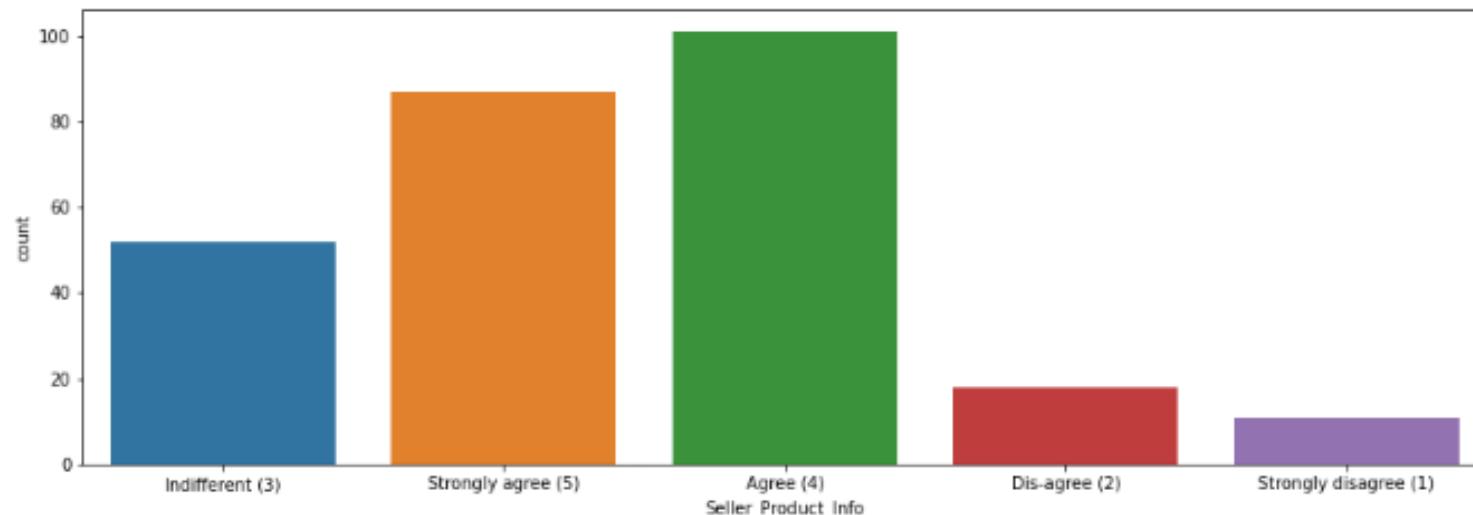
Information on similar product to the one highlighted is important for product comparison is Strongly agree most and got rating 5 for it from Total no of customers = 116 and Dis-agree got rating 2 from only total no of 18 customers.

Data Visualization : Based on Seller Product Info

```
In [41]: #Count Plot for "Seller_Product_Info" column  
print(retention[\"Seller_Product_Info\"].value_counts())  
plt.figure(figsize=(15,5))  
sns.countplot("Seller_Product_Info",data=retention)
```

```
Agree (4)          101  
Strongly agree (5) 87  
Indifferent (3)    52  
Dis-agree (2)      18  
Strongly disagree (1) 11  
Name: Seller_Product_Info, dtype: int64
```

```
Out[41]: <AxesSubplot:xlabel='Seller_Product_Info', ylabel='count'>
```



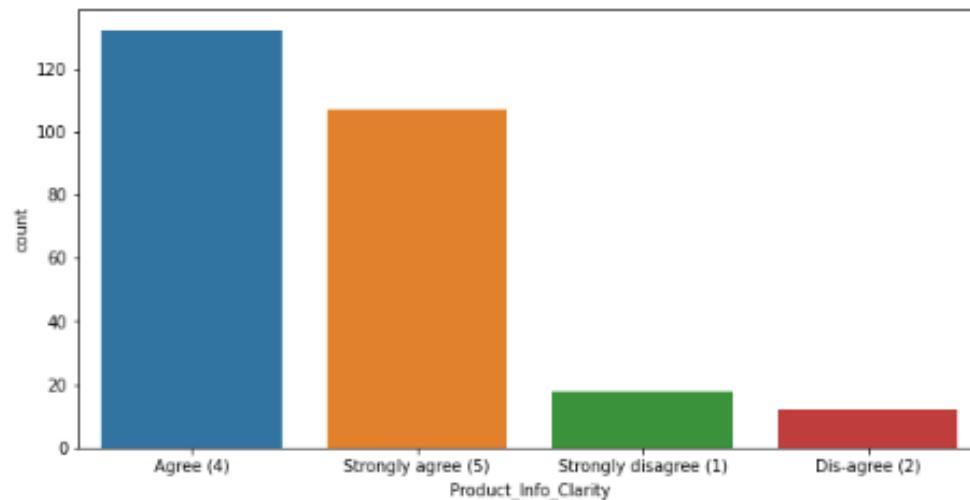
Complete information on listed seller and product being offered is important for purchase decision is Agree most and got rating 4 for it from Total no of customers = 101 and Strongly disagree got rating 1 from only total no of 11 customers.

Data Visualization : Based on Product Info Clarity

```
In [42]: #Count Plot for "Product_Info_Clarity" column  
print(retention["Product_Info_Clarity"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Product_Info_Clarity",data=retention)
```

```
Agree (4)           132  
Strongly agree (5) 107  
Strongly disagree (1) 18  
Dis-agree (2)       12  
Name: Product_Info_Clarity, dtype: int64
```

```
Out[42]: <AxesSubplot:xlabel='Product_Info_Clarity', ylabel='count'>
```



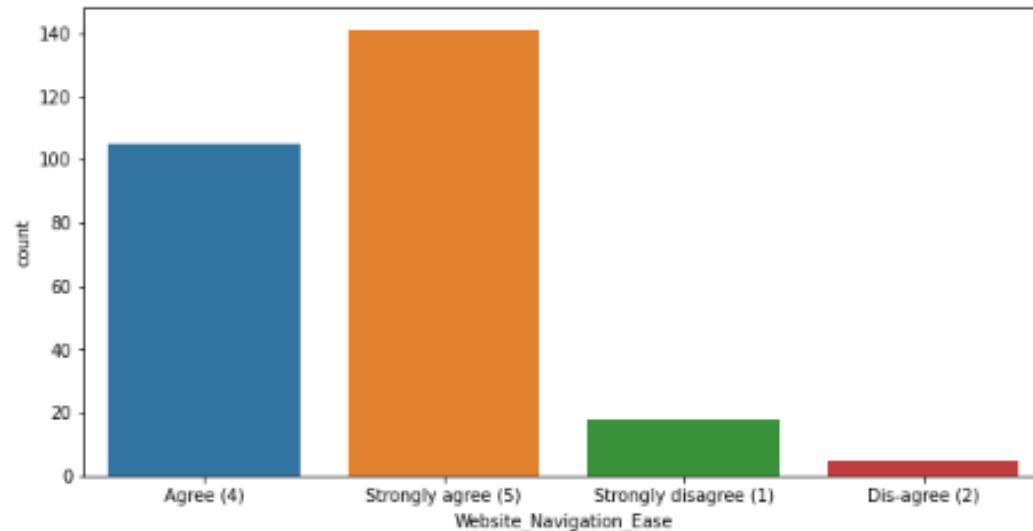
All relevant information on listed products must be stated clearly is Agree most and got rating 4 from Total no of customers = 132 and Dis-agree got rating 2 from only total no of 12 customers.

Data Visualization : Based on Website Navigation ease

```
In [43]: #Count Plot for "Website_Navigation_Ease" column  
print(retention["Website_Navigation_Ease"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Website_Navigation_Ease",data=retention)
```

```
Strongly agree (5)      141  
Agree (4)              105  
Strongly disagree (1)  18  
Dis-agree (2)           5  
Name: Website_Navigation_Ease, dtype: int64
```

```
Out[43]: <AxesSubplot:xlabel='Website_Navigation_Ease', ylabel='count'>
```



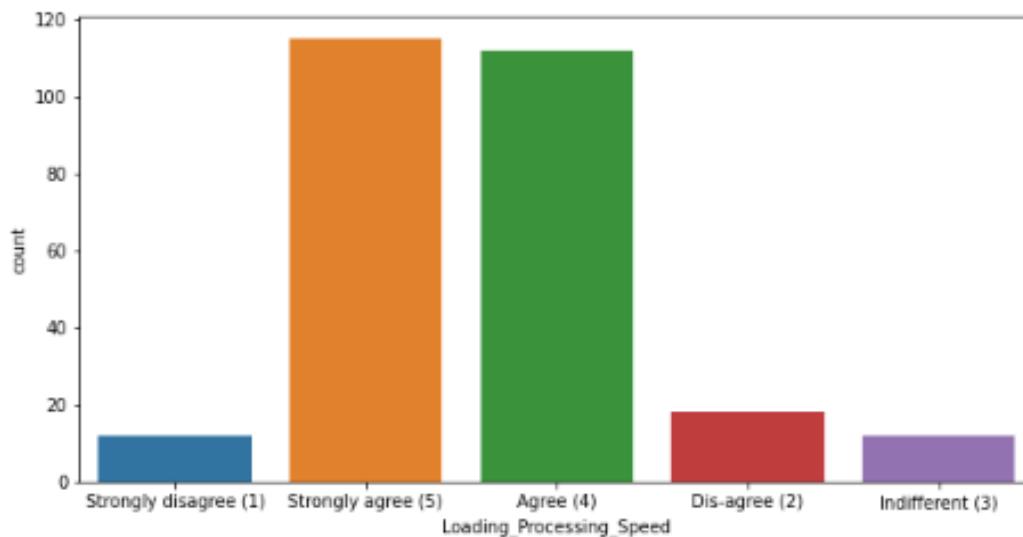
Ease of navigation in website is Strongly agree most and got rating 5 from Total no of customers = 141 and Dis-agree got rating 2 from only total no of 5 customers.

Data Visualization : Based on Loading speed

```
In [44]: #Count Plot for "Loading_Processing_Speed" column  
print(retention[ "Loading_Processing_Speed"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Loading_Processing_Speed",data=retention)
```

```
Strongly agree (5)      115  
Agree (4)              112  
Dis-agree (2)           18  
Strongly disagree (1)   12  
Indifferent (3)         12  
Name: Loading_Processing_Speed, dtype: int64
```

```
out[44]: <AxesSubplot:xlabel='Loading_Processing_Speed', ylabel='count'>
```



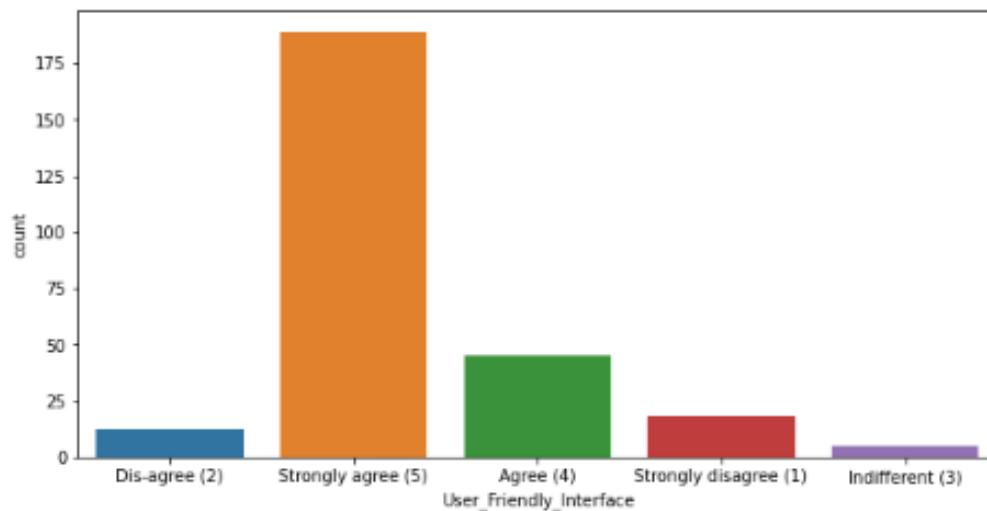
Loading and processing speed is Strongly agree most and got rating 5 from Total no of customers = 115 and Indifferent got rating 3 from only total no of 12 customers.

Data Visualization : Based on User Friendly Interface

```
In [45]: #Count Plot for "User_Friendly_Interface" column  
print(retention["User_Friendly_Interface"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("User_Friendly_Interface",data=retention)
```

```
Strongly agree (5)      189  
Agree (4)              45  
Strongly disagree (1)  18  
Dis-agree (2)           12  
Indifferent (3)         5  
Name: User_Friendly_Interface, dtype: int64
```

```
Out[45]: <AxesSubplot:xlabel='User_Friendly_Interface', ylabel='count'>
```



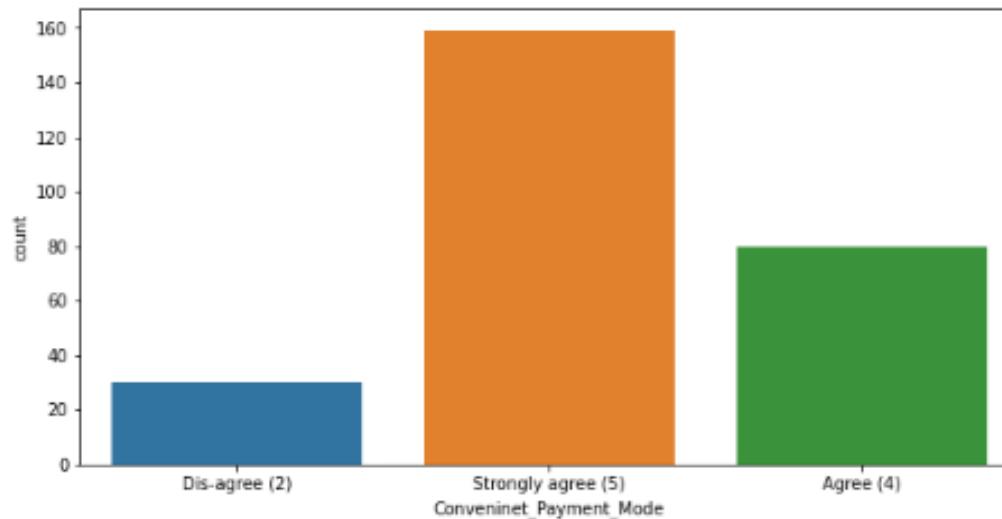
User friendly Interface of the website is Strongly agree most and got rating 5 from Total no of customers 189 and Indifferent got rating 3 from only total no of 5 customers.

Data Visualization : Based on Convenient Payment mode

```
In [46]: #Count Plot for "Conveninet_Payment_Mode" column  
print(retention["Conveninet_Payment_Mode"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Conveninet_Payment_Mode",data=retention)
```

```
Strongly agree (5)    159  
Agree (4)            80  
Dis-agree (2)         30  
Name: Conveninet_Payment_Mode, dtype: int64
```

```
Out[46]: <AxesSubplot:xlabel='Conveninet_Payment_Mode', ylabel='count'>
```



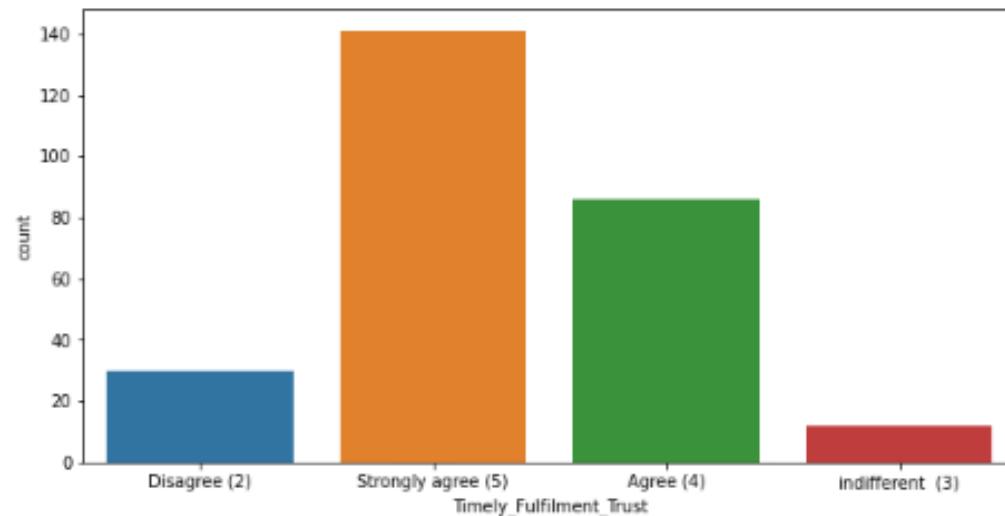
Convenient Payment methods is Strongly agree most and got rating 5 from Total no of customers 159 and Dis-agree got rating 2 from only total no of 30 customers.

Data Visualization : Based on Timely Fulfillment trust

```
In [47]: #Count Plot for "Timely_Fulfilment_Trust" column  
print(retention["Timely_Fulfilment_Trust"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Timely_Fulfilment_Trust",data=retention)
```

```
Strongly agree (5)    141  
Agree (4)           86  
Disagree (2)         30  
indifferent (3)      12  
Name: Timely_Fulfilment_Trust, dtype: int64
```

```
Out[47]: <AxesSubplot:xlabel='Timely_Fulfilment_Trust', ylabel='count'>
```



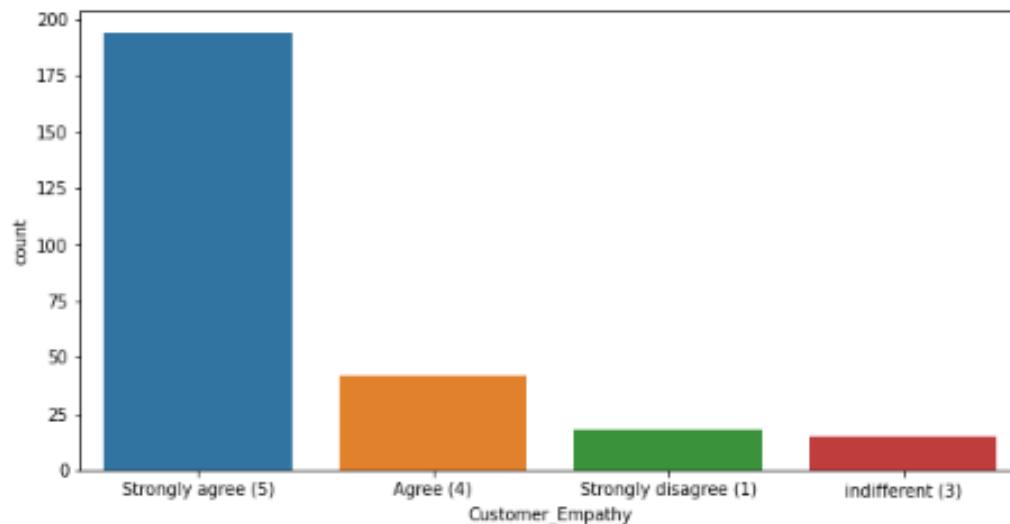
Trust that the online retail store will fulfill its part of the transaction at the stipulated time is Strongly agree most and got rating 5 from Total no of customers 141 and indifferent got rating 3 from only total no of 12 customers.

Data Visualization : Based on Customer Empathy

```
In [48]: #Count Plot for "Customer_Empathy" column  
print(retention["Customer_Empathy"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Customer_Empathy",data=retention)
```

```
Strongly agree (5)      194  
Agree (4)              42  
Strongly disagree (1)  18  
indifferent (3)        15  
Name: Customer_Empathy, dtype: int64
```

```
Out[48]: <AxesSubplot:xlabel='Customer_Empathy', ylabel='count'>
```



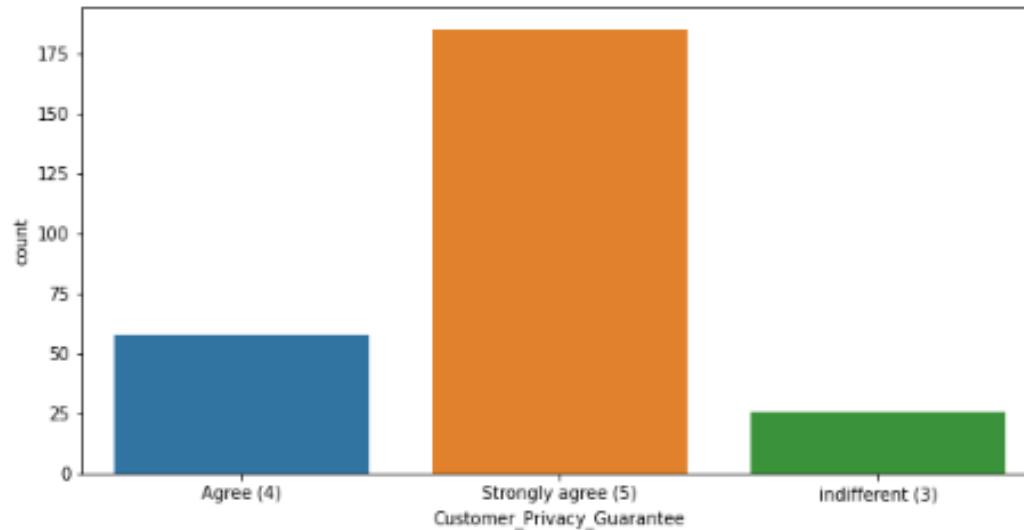
Empathy (readiness to assist with queries) towards the customers is Strongly agree most and got rating 5 from Total no of customers 194 and indifferent got rating 3 from only total no of 15 customers.

Data Visualization : Based on Customer Privacy Guarantee

```
In [49]: #Count Plot for "Customer_Privacy_Guarantee" column  
print(retention["Customer_Privacy_Guarantee"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Customer_Privacy_Guarantee",data=retention)
```

```
Strongly agree (5)    185  
Agree (4)           58  
indifferent (3)     26  
Name: Customer_Privacy_Guarantee, dtype: int64
```

```
Out[49]: <AxesSubplot:xlabel='Customer_Privacy_Guarantee', ylabel='count'>
```



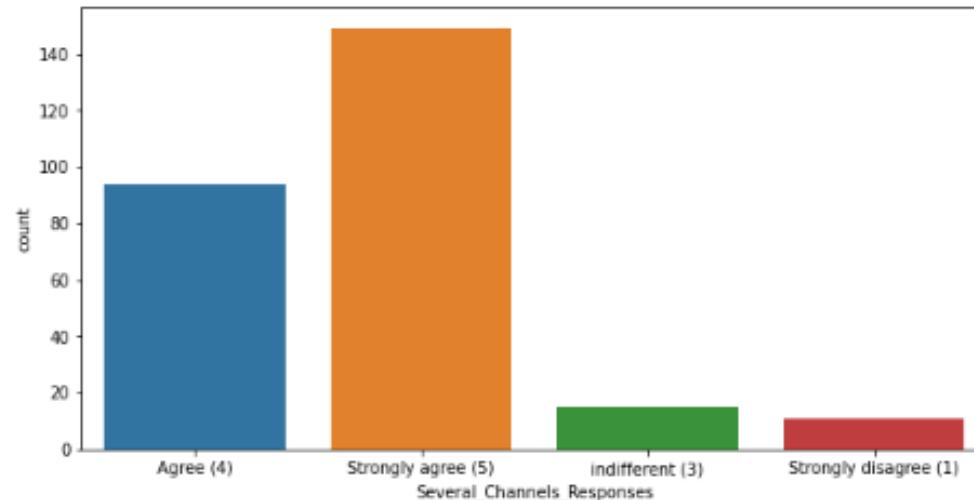
Being able to guarantee the privacy of the customer is Strongly agree most and got rating 5 from Total no of customers 185 and indifferent got rating 3 from only total no of 26 customers.

Data Visualization : Based on Several Channel Response

```
In [50]: #Count Plot for "Several_Channels_Responses" column  
print(retention["Several_Channels_Responses"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Several_Channels_Responses",data=retention)
```

```
Strongly agree (5)      149  
Agree (4)              94  
indifferent (3)        15  
Strongly disagree (1)  11  
Name: Several_Channels_Responses, dtype: int64
```

```
Out[50]: <AxesSubplot:xlabel='Several_Channels_Responses', ylabel='count'>
```



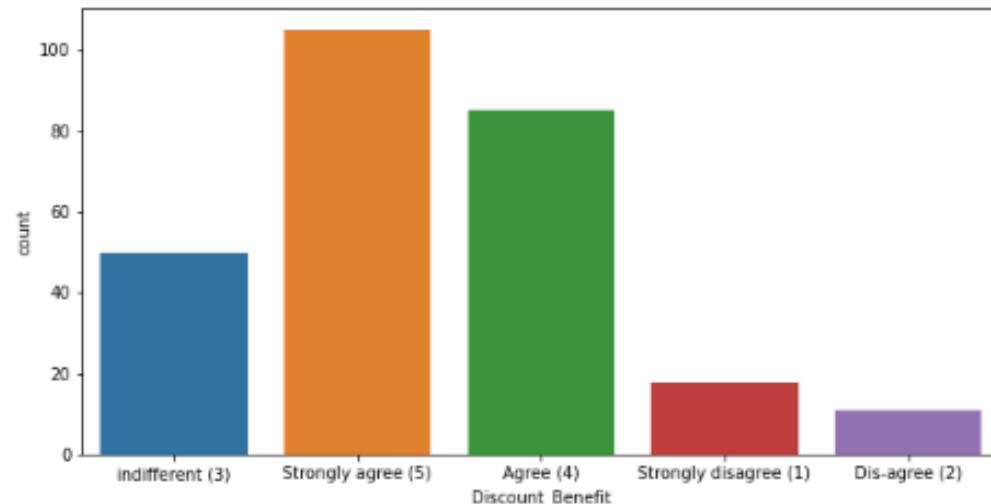
Responsiveness, availability of several communication channels (email, online rep, twitter, phone etc.) is Strongly agree most and got rating 5 from Total no of customers 149 and Strongly disagree got rating 1 from only total no of 11 customers.

Data Visualization : Based on Discount Benefit

```
In [51]: #Count Plot for "Discount_Benefit" column  
print(retention["Discount_Benefit"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Discount_Benefit",data=retention)
```

```
Strongly agree (5)      105  
Agree (4)              85  
indifferent (3)        50  
Strongly disagree (1)   18  
Dis-agree (2)           11  
Name: Discount_Benefit, dtype: int64
```

```
Out[51]: <AxesSubplot:xlabel='Discount_Benefit', ylabel='count'>
```



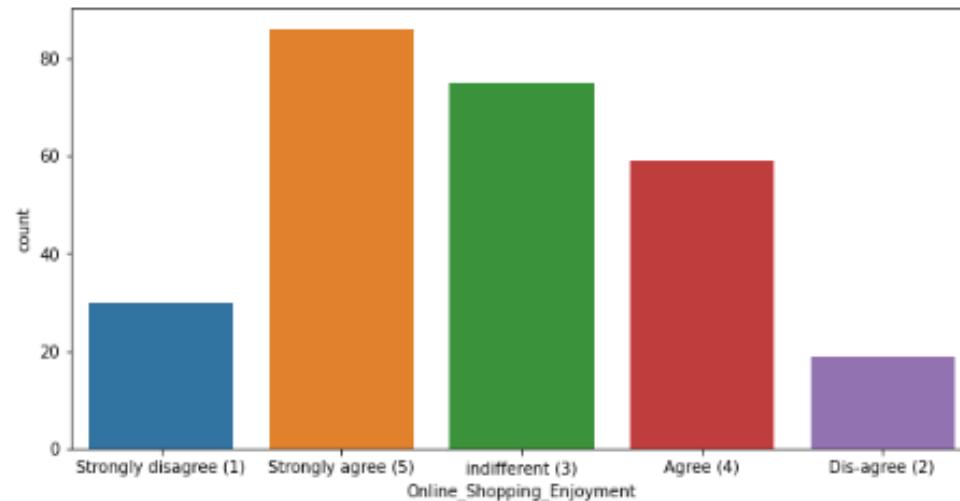
Online shopping gives monetary benefit and discounts is Strongly agree most and got rating 5 from Total no of customers 105 and Dis-agree got rating 2 from only total no of 11 customers.

Data Visualization : Based on Online Shopping Enjoyment

```
In [52]: #Count Plot for "Online_Shopping_Enjoyment" column  
print(retention["Online_Shopping_Enjoyment"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Online_Shopping_Enjoyment",data=retention)
```

```
Strongly agree (5)      86  
indifferent (3)        75  
Agree (4)              59  
Strongly disagree (1)  30  
Dis-agree (2)          19  
Name: Online_Shopping_Enjoyment, dtype: int64
```

```
Out[52]: <AxesSubplot:xlabel='Online_Shopping_Enjoyment', ylabel='count'>
```



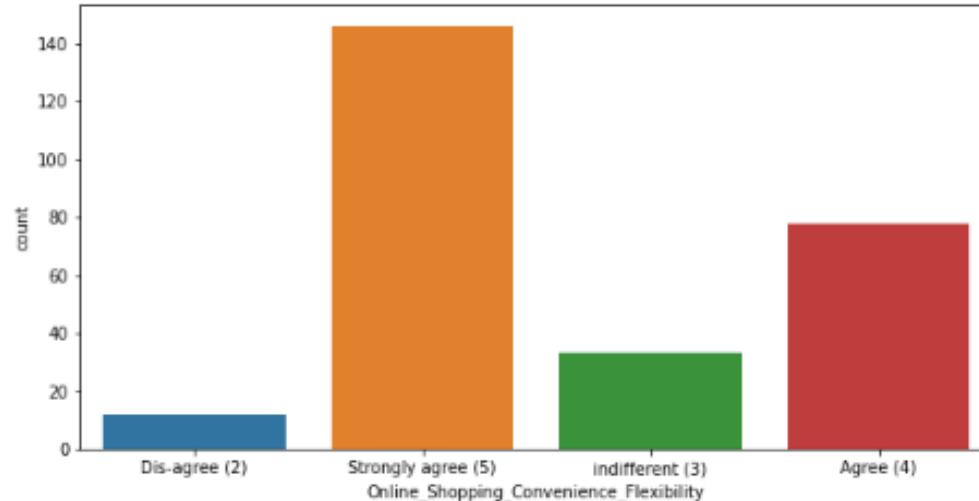
Enjoyment is derived from shopping online is Strongly agree most and got rating 5 from Total no of customers 86 and Dis-agree got rating 2 from only total no of 19 customers.

Data Visualization : Based on Online Shopping Convenience Flexibility

```
In [53]: #Count Plot for "Online_Shopping_Convenience_Flexibility" column  
print(retention["Online_Shopping_Convenience_Flexibility"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Online_Shopping_Convenience_Flexibility",data=retention)
```

```
Strongly agree (5)    146  
Agree (4)            78  
Indifferent (3)      33  
Dis-agree (2)         12  
Name: Online_Shopping_Convenience_Flexibility, dtype: int64
```

```
Out[53]: <AxesSubplot:xlabel='Online_Shopping_Convenience_Flexibility', ylabel='count'>
```



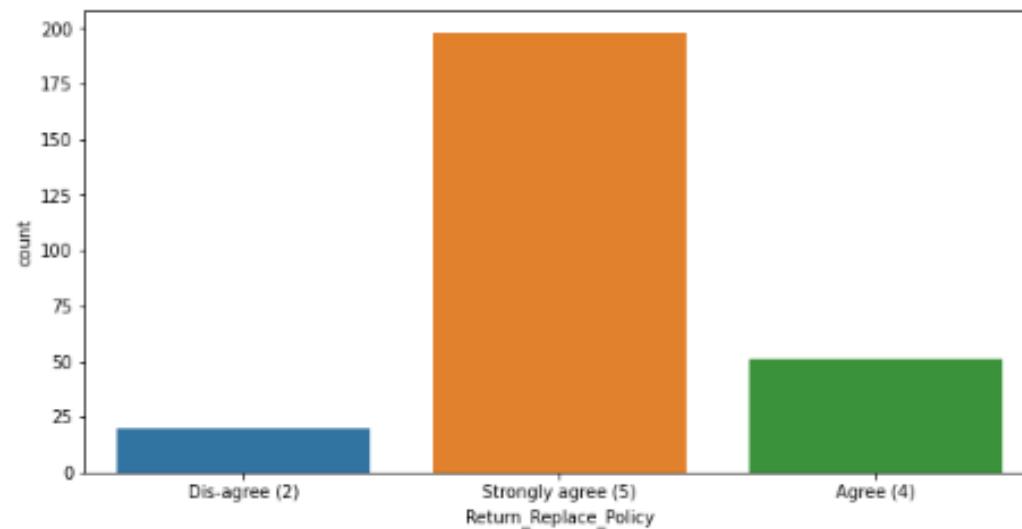
Shopping online is convenient and flexible is Strongly agree most and got rating 5 from Total no of customers 146 and Dis-agree got rating 2 from only total no of 12 customers.

Data Visualization : Based on Return Replace Policy

```
In [54]: #Count Plot for "Return_Replace_Policy" column  
print(retention["Return_Replace_Policy"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Return_Replace_Policy",data=retention)
```

```
Strongly agree (5)    198  
Agree (4)           51  
Dis-agree (2)        20  
Name: Return_Replace_Policy, dtype: int64
```

```
Out[54]: <AxesSubplot:xlabel='Return_Replace_Policy', ylabel='count'>
```



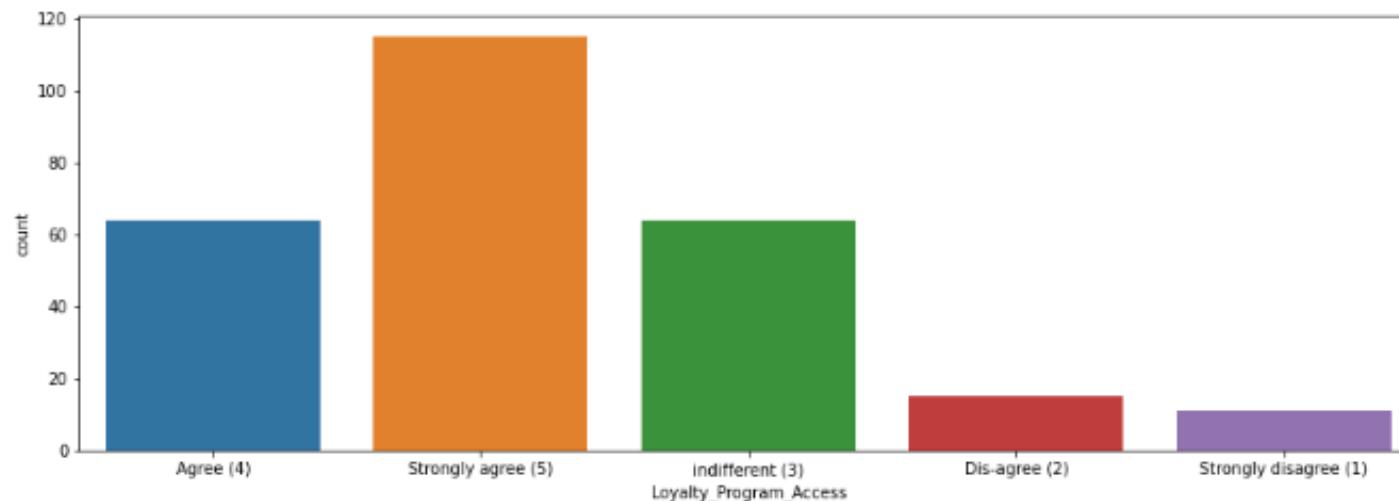
Return and replacement policy of the e-tailer is important for purchase decision is Strongly agree most and got rating 5 from Total no of customers 198 and Dis-agree got rating 2 from only total no of 20 customers.

Data Visualization : Based on Loyalty Program Access

```
In [55]: #Count Plot for "Loyalty_Program_Access" column  
print(retention["Loyalty_Program_Access"].value_counts())  
plt.figure(figsize=(15,5))  
sns.countplot("Loyalty_Program_Access",data=retention)
```

```
Strongly agree (5)      115  
Agree (4)              64  
indifferent (3)        64  
Dis-agree (2)           15  
Strongly disagree (1)   11  
Name: Loyalty_Program_Access, dtype: int64
```

```
Out[55]: <AxesSubplot:xlabel='Loyalty_Program_Access', ylabel='count'>
```



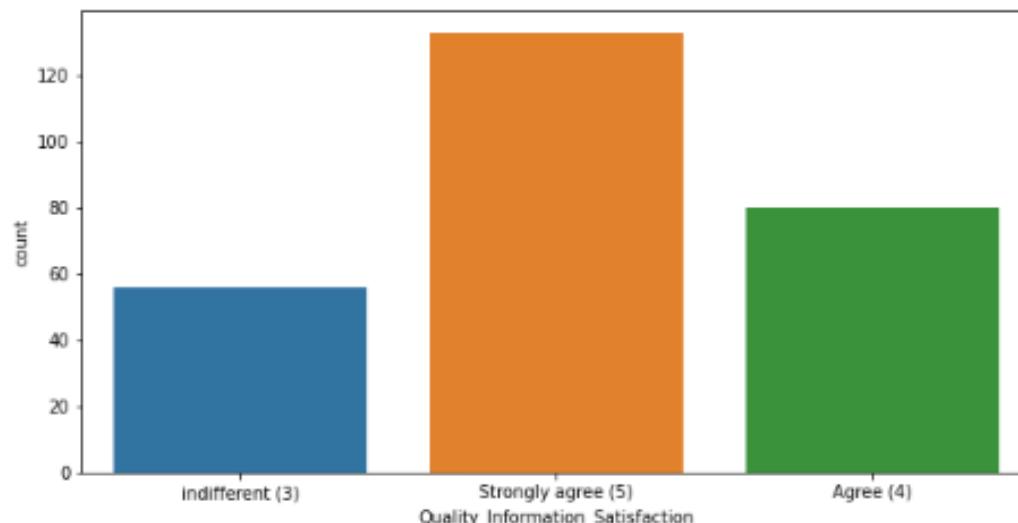
Gaining access to loyalty programs is a benefit of shopping online is Strongly agree most and got rating 5 from Total no of customers 115 and Strongly disagree got rating 1 from only total no of 11 customers.

Data Visualization : Based on Quality Information Satisfaction

```
In [56]: #Count Plot for "Quality_Information_Satisfaction" column  
print(retention["Quality_Information_Satisfaction"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Quality_Information_Satisfaction",data=retention)
```

```
Strongly agree (5)      133  
Agree (4)              80  
Indifferent (3)        56  
Name: Quality_Information_Satisfaction, dtype: int64
```

```
Out[56]: <AxesSubplot:xlabel='Quality_Information_Satisfaction', ylabel='count'>
```



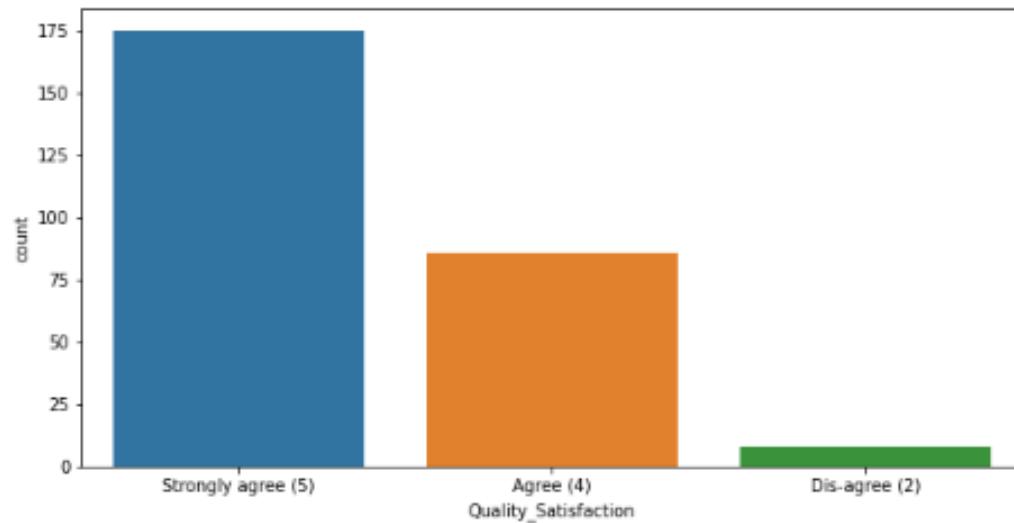
Displaying quality Information on the website improves satisfaction of customers is Strongly agree most and got rating 5 from Total no of customers 133 and indifferent got rating 3 from only total no of 56 customers.

Data Visualization : Based on Quality Satisfaction

```
In [57]: #Count Plot for "Quality_Satisfaction" column  
print(retention["Quality_Satisfaction"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Quality_Satisfaction",data=retention)
```

```
Strongly agree (5)    175  
Agree (4)           86  
Dis-agree (2)        8  
Name: Quality_Satisfaction, dtype: int64
```

```
Out[57]: <AxesSubplot:xlabel='Quality_Satisfaction', ylabel='count'>
```



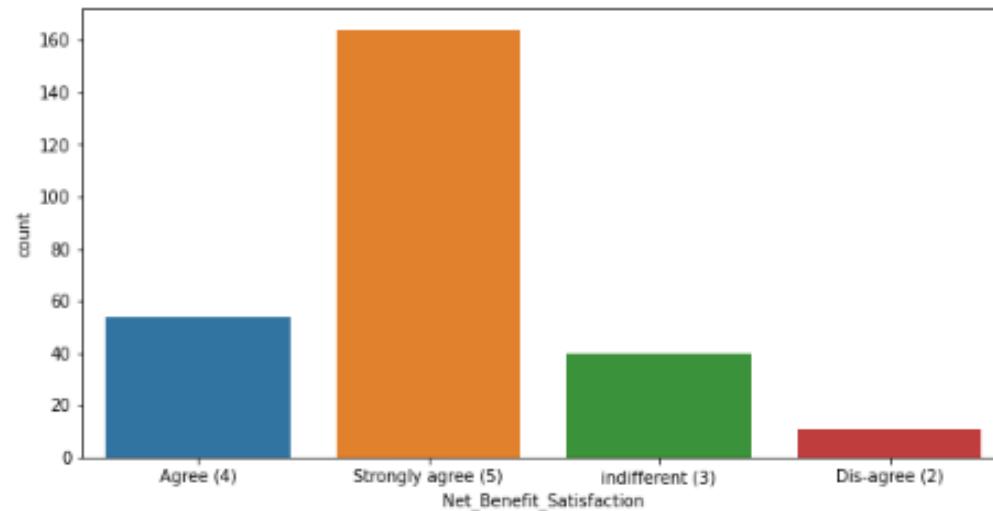
User derive satisfaction while shopping on a good quality website or application is Strongly agree most and got rating 5 from Total no of customers 175 and Dis-agree got rating 2 from only total no of 8 customers.

Data Visualization : Based on Net Benefit Satisfaction

```
In [58]: #Count Plot for "Net_Benefit_Satisfaction" column  
print(retention["Net_Benefit_Satisfaction"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Net_Benefit_Satisfaction",data=retention)
```

```
Strongly agree (5)    164  
Agree (4)           54  
indifferent (3)     40  
Dis-agree (2)        11  
Name: Net_Benefit_Satisfaction, dtype: int64
```

```
Out[58]: <AxesSubplot:xlabel='Net_Benefit_Satisfaction', ylabel='count'>
```



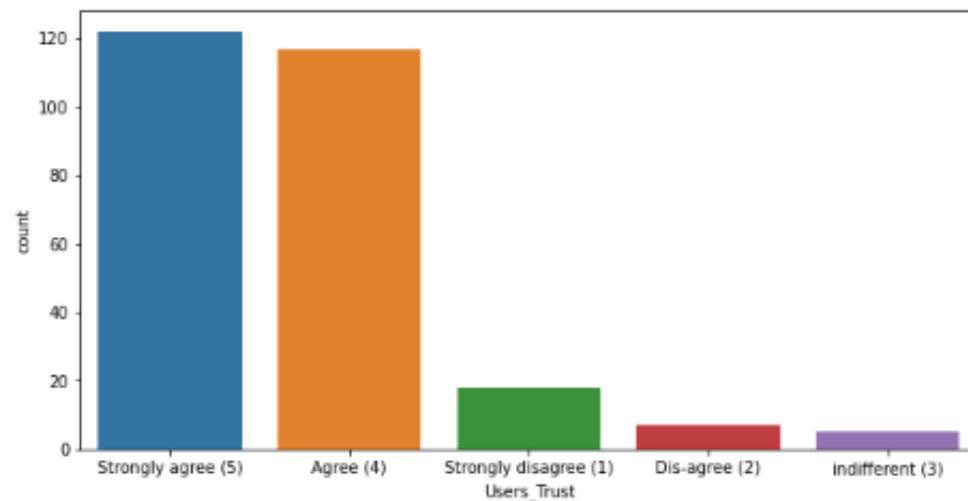
Net Benefit derived from shopping online can lead to users satisfaction is Strongly agree most and got rating 5 from Total no of customers 164 and Dis-agree got rating 2 from only total no of 11 customers.

Data Visualization : Based on Users Trust

```
In [59]: #Count Plot for "Users_Trust" column
print(retention["Users_Trust"].value_counts())
plt.figure(figsize=(10,5))
sns.countplot("Users_Trust",data=retention)
```

```
Strongly agree (5)      122
Agree (4)              117
Strongly disagree (1)  18
Dis-agree (2)           7
indifferent (3)        5
Name: Users_Trust, dtype: int64
```

```
Out[59]: <AxesSubplot:xlabel='Users_Trust', ylabel='count'>
```



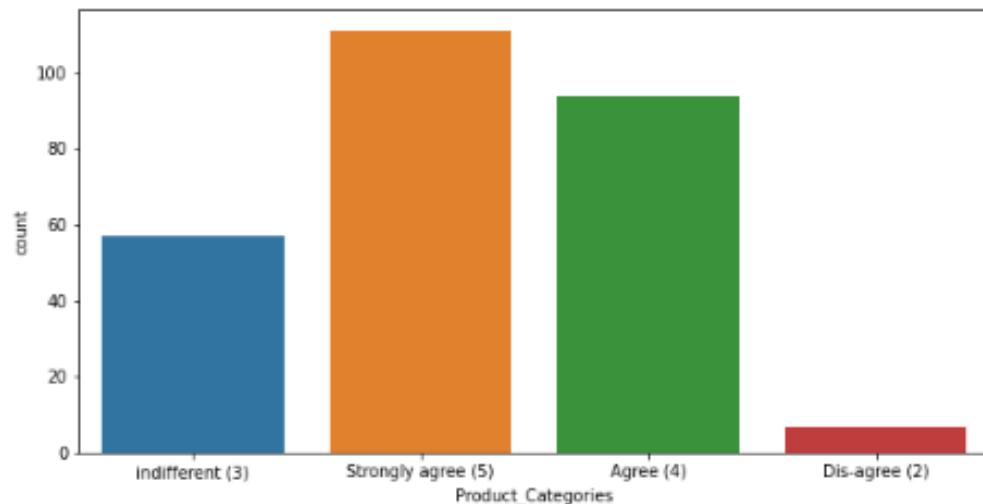
User satisfaction cannot exist without trust is Strongly agree most and got rating 5 from Total no of customers 122 and indifferent got rating 3 from only total no of 5 customers.

Data Visualization : Based on Product Categories

```
In [60]: #Count Plot for "Product_Categories" column  
print(retention["Product_Categories"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Product_Categories",data=retention)
```

```
Strongly agree (5)    111  
Agree (4)           94  
indifferent (3)     57  
Dis-agree (2)        7  
Name: Product_Categories, dtype: int64
```

```
Out[60]: <AxesSubplot:xlabel='Product_Categories', ylabel='count'>
```



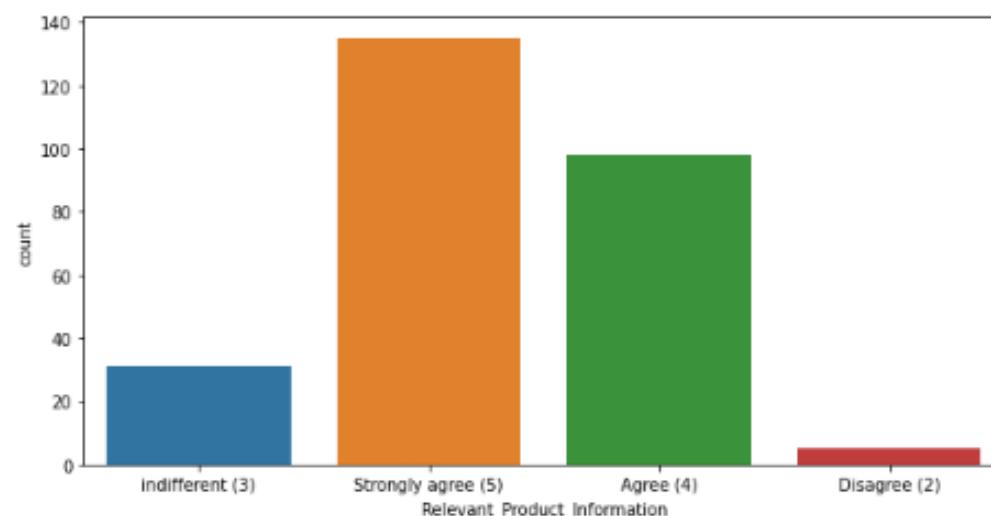
Offering a wide variety of listed product in several category is Strongly agree most and got rating 5 from Total no of customers 111 and Dis-agree got rating 2 from only total no of 7 customers.

Data Visualization : Based on Relevant product Information

```
In [61]: #Count Plot for "Relevant_Product_Information" column  
print(retention["Relevant_Product_Information"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Relevant_Product_Information",data=retention)
```

```
Strongly agree (5)    135  
Agree (4)           98  
indifferent (3)     31  
Disagree (2)         5  
Name: Relevant_Product_Information, dtype: int64
```

```
Out[61]: <AxesSubplot:xlabel='Relevant_Product_Information', ylabel='count'>
```



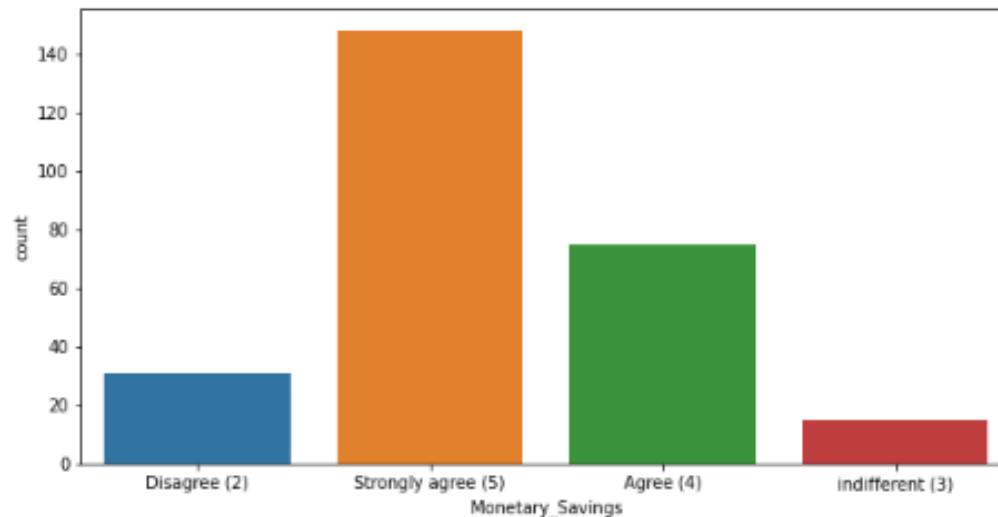
Provision of complete and relevant product information is Strongly agree most and got rating 5 from Total no of customers 135 and Disagree got rating 2 from only total no of 5 customers.

Data Visualization : Based on Monetary Savings

```
In [62]: #Count Plot for "Monetary_Savings" column
print(retention["Monetary_Savings"].value_counts())
plt.figure(figsize=(10,5))
sns.countplot("Monetary_Savings",data=retention)
```

```
Strongly agree (5)    148
Agree (4)            75
Disagree (2)          31
indifferent (3)       15
Name: Monetary_Savings, dtype: int64
```

```
Out[62]: <AxesSubplot:xlabel='Monetary_Savings', ylabel='count'>
```



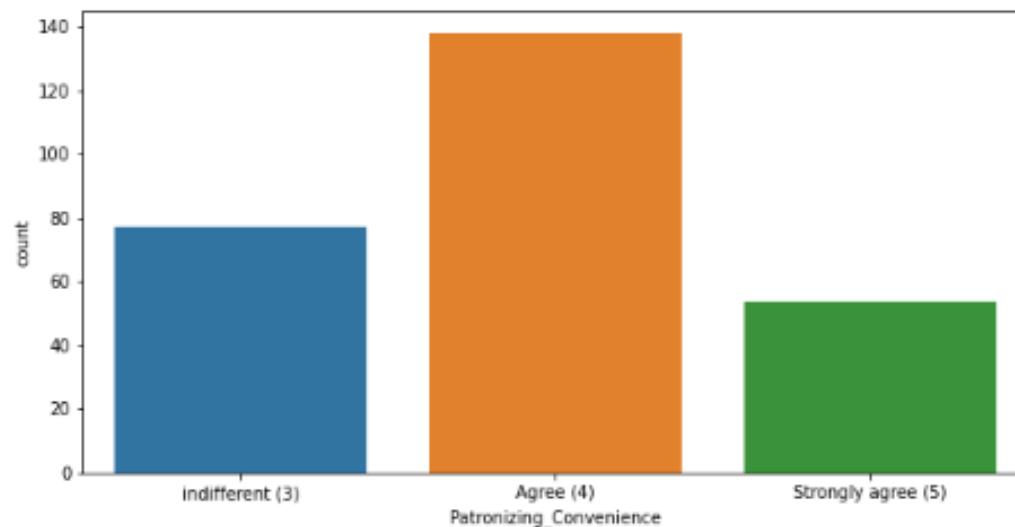
Monetary savings is Strongly agree most and got rating 5 from Total no of customers 148 and indifferent got rating 3 from only total no of 15 customers.

Data Visualization : Based on Patronizing Convenience

```
In [63]: #Count Plot for "Patronizing_Convenience" column
print(retention["Patronizing_Convenience"].value_counts())
plt.figure(figsize=(10,5))
sns.countplot("Patronizing_Convenience",data=retention)
```

```
Agree (4)      138
indifferent (3) 77
Strongly agree (5) 54
Name: Patronizing_Convenience, dtype: int64
```

```
Out[63]: <AxesSubplot:xlabel='Patronizing_Convenience', ylabel='count'>
```



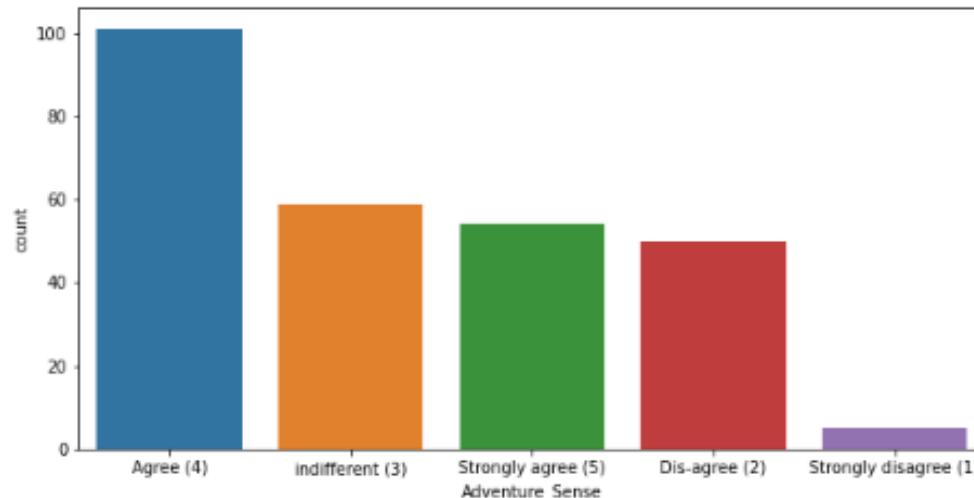
The Convenience of patronizing the online retailer is Agree most and got rating 4 from Total no of customers 138 and Strongly agree got rating 5 from only total no of 54 customers.

Data Visualization : Based on Adventure Sense

```
In [64]: #Count Plot for "Adventure_Sense" column  
print(retention["Adventure_Sense"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Adventure_Sense",data=retention)
```

```
Agree (4)          101  
indifferent (3)    59  
Strongly agree (5) 54  
Dis-agree (2)       50  
Strongly disagree (1) 5  
Name: Adventure_Sense, dtype: int64
```

```
Out[64]: <AxesSubplot:xlabel='Adventure_Sense', ylabel='count'>
```



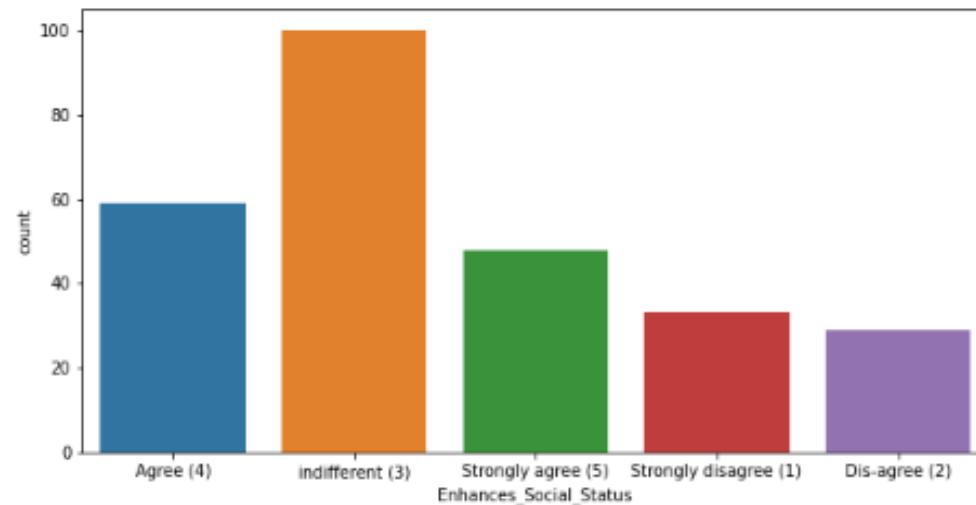
Shopping on the website gives you the sense of adventure is Agree most and got rating 4 from Total no of customers 101 and Strongly disagree got rating 1 from only total no of 5 customers.

Data Visualization : Based on Enhances Social status

```
In [65]: #Count Plot for "Enhances_Social_Status" column
print(retention["Enhances_Social_Status"].value_counts())
plt.figure(figsize=(10,5))
sns.countplot("Enhances_Social_Status",data=retention)
```

```
indifferent (3)      100
Agree (4)           59
Strongly agree (5)  48
Strongly disagree (1) 33
Dis-agree (2)        29
Name: Enhances_Social_Status, dtype: int64
```

```
Out[65]: <AxesSubplot:xlabel='Enhances_Social_Status', ylabel='count'>
```



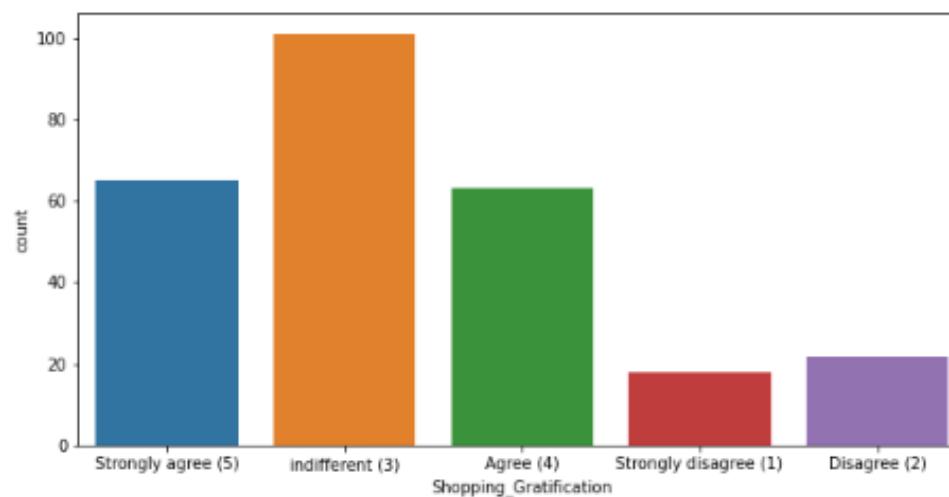
Shopping on your preferred e-tailer enhances your social status is indifferent most and got rating 3 from Total no of customers 100 and Dis-agree got rating 2 from only total no of 29 customers.

Data Visualization : Based on Shopping Gratification

```
In [66]: #Count Plot for "Users_Trust" column  
print(retention[\"Shopping_Gratification\"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot(\"Shopping_Gratification\",data=retention)
```

```
indifferent (3)      101  
Strongly agree (5)   65  
Agree (4)           63  
Disagree (2)         22  
Strongly disagree (1) 18  
Name: Shopping_Gratification, dtype: int64
```

```
Out[66]: <AxesSubplot:xlabel='Shopping_Gratification', ylabel='count'>
```



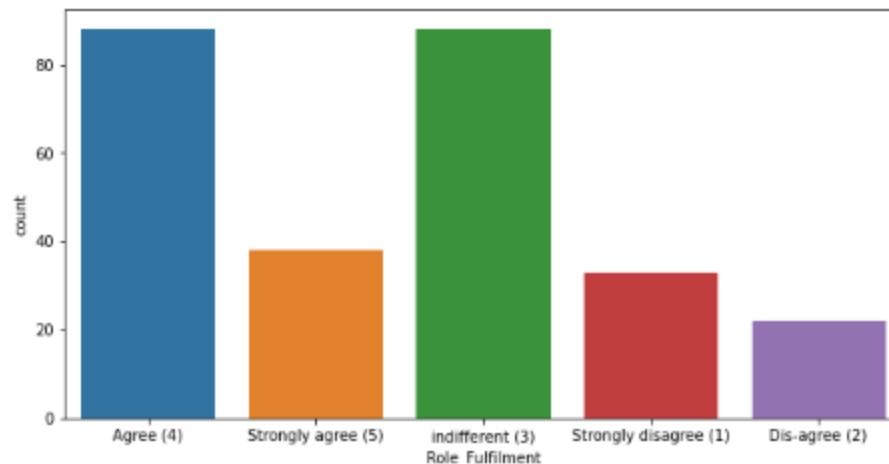
You feel gratification shopping on your favorite e-tailer is indifferent most and got rating 3 from Total no of customers 101 and Strongly disagree got rating 1 from only total no of 18 customers.

Data Visualization : Based on Role Fulfillment

```
In [67]: #Count Plot for "Role_Fulfilment" column  
print(retention["Role_Fulfilment"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Role_Fulfilment",data=retention)
```

```
Agree (4)      88  
indifferent (3) 88  
Strongly agree (5) 38  
Strongly disagree (1) 33  
Dis-agree (2) 22  
Name: Role_Fulfilment, dtype: int64
```

```
Out[67]: <AxesSubplot:xlabel='Role_Fulfilment', ylabel='count'>
```



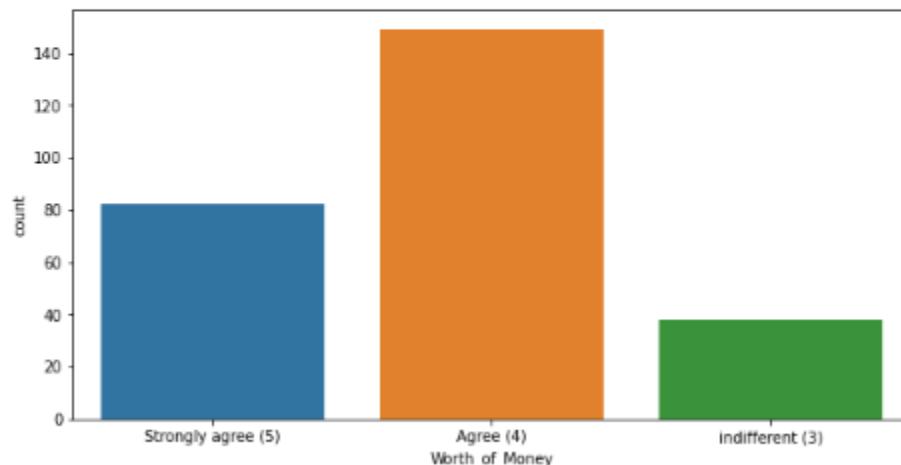
Shopping on the website helps you fulfill certain roles is Agree most and got rating 4 from Total no of customers 88 and Dis-agree got rating 2 from only total no of 22 customers.

Data Visualization : Based on Worth of Money

```
In [68]: #Count Plot for "Worth_of_Money" column
print(retention["Worth_of_Money"].value_counts())
plt.figure(figsize=(10,5))
sns.countplot("Worth_of_Money",data=retention)
```

```
Agree (4)          149
Strongly agree (5) 82
indifferent (3)    38
Name: Worth_of_Money, dtype: int64
```

```
out[68]: <AxesSubplot:xlabel='Worth_of_Money', ylabel='count'>
```



Getting value for money spent is Agree most and got rating 4 from Total no of customers 149 and indifferent got rating 3 from only total no of 38 customers.

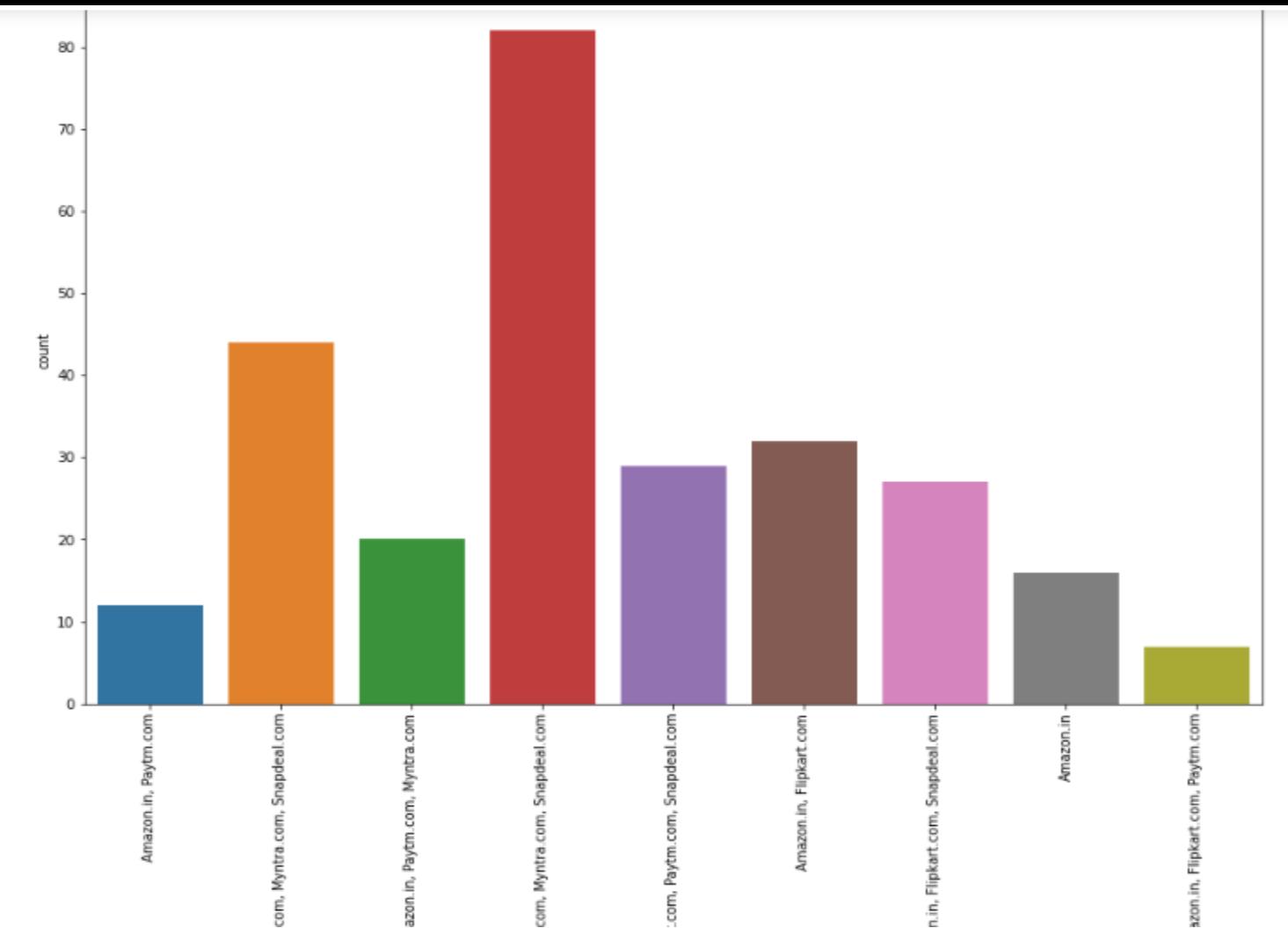
Data Visualization : Based on Shopped from Retailer

```
In [69]: #Count Plot for "Shopped_From_Retailer" column
print(retention["Shopped_From_Retailer"].value_counts())
plt.figure(figsize=(15,10))
sns.countplot("Shopped_From_Retailer",data=retention)
plt.xticks(rotation=90)
```

```
Amazon.in, Flipkart.com, Paytm.com, Myntra.com, Snapdeal.com    82
Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com                44
Amazon.in, Flipkart.com                                         32
Amazon.in, Flipkart.com, Paytm.com, Snapdeal.com                29
Amazon.in, Flipkart.com, Snapdeal.com                           27
Amazon.in, Paytm.com, Myntra.com                                20
Amazon.in                                                       16
Amazon.in, Paytm.com                                           12
Amazon.in, Flipkart.com, Paytm.com                               7
Name: Shopped_From_Retailer, dtype: int64
```

```
Out[69]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8]),
 [Text(0, 0, 'Amazon.in, Paytm.com'),
  Text(1, 0, 'Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com'),
  Text(2, 0, 'Amazon.in, Paytm.com, Myntra.com'),
  Text(3, 0, 'Amazon.in, Flipkart.com, Paytm.com, Myntra.com, Snapdeal.com'),
  Text(4, 0, 'Amazon.in, Flipkart.com, Paytm.com, Snapdeal.com'),
  Text(5, 0, 'Amazon.in, Flipkart.com'),
  Text(6, 0, 'Amazon.in, Flipkart.com, Snapdeal.com'),
  Text(7, 0, 'Amazon.in'),
  Text(8, 0, 'Amazon.in, Flipkart.com, Paytm.com')])
```

Data Visualization : Based on Shopped from Retailer



From the following, tick any (or all) of the online retailers you have shopped from Amazon.in, Flipkart.com, Paytm.com, Myntra.com, Snapdeal.com most (Total No= 82) most and from Amazon.in, Flipkart.com, Paytm.com (Total No= 7) least.

Data Visualization for Ease Website Application

```
In [70]: #Count Plot for "Ease_Website_Application" column
print(retention[ "Ease_Website_Application"].value_counts())
plt.figure(figsize=(15,5))
sns.countplot("Ease_Website_Application",data=retention)
plt.xticks(rotation=90)

Amazon.in, Flipkart.com, Paytm.com, Myntra.com, Snapdeal.com      64
Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com                  44
Amazon.in, Flipkart.com                                           44
Amazon.in                                                       29
Amazon.in, Flipkart.com, Paytm.com, Snapdeal.com                 22
Amazon.in, Paytm.com, Myntra.com                                 20
Amazon.in, Flipkart.com, Myntra.com                               19
Paytm.com                                                       12
Flipkart.com                                                    8
Amazon.in, Paytm.com                                            7
Name: Ease_Website_Application, dtype: int64

Out[70]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
          [Text(0, 0, 'Paytm.com'),
           Text(1, 0, 'Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com'),
           Text(2, 0, 'Amazon.in, Paytm.com, Myntra.com'),
           Text(3, 0, 'Amazon.in, Flipkart.com, Paytm.com, Myntra.com, Snapdeal.com'),
           Text(4, 0, 'Amazon.in, Flipkart.com, Paytm.com, Snapdeal.com'),
           Text(5, 0, 'Amazon.in, Flipkart.com'),
           Text(6, 0, 'Amazon.in, Flipkart.com, Myntra.com'),
           Text(7, 0, 'Amazon.in'),
           Text(8, 0, 'Amazon.in, Paytm.com'),
           Text(9, 0, 'Flipkart.com')])
```

Data Visualization for Visually Appealing Layout

```
In [71]: #Count Plot for "Visually_Appealing_Layout" column
print(retention["Visually_Appealing_layout"].value_counts())
plt.figure(figsize=(10,5))
sns.countplot("Visually_Appealing_layout",data=retention)
plt.xticks(rotation=90)
```

Amazon.in, Flipkart.com	87
Amazon.in	44
Amazon.in, Flipkart.com, Paytm.com, Myntra.com, Snapdeal.com	36
Amazon.in, Paytm.com, Myntra.com	20
Amazon.in, Myntra.com	15
Myntra.com	15
Flipkart.com, Myntra.com	15
Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com	14
Flipkart.com	12
Amazon.in, Flipkart.com, Paytm.com, Snapdeal.com	11
Name: Visually_Appealing_layout, dtype: int64	

```
Out[71]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
 [Text(0, 0, 'Flipkart.com'),
  Text(1, 0, 'Amazon.in, Myntra.com'),
  Text(2, 0, 'Amazon.in, Paytm.com, Myntra.com'),
  Text(3, 0, 'Amazon.in, Flipkart.com, Paytm.com, Myntra.com, Snapdeal.com'),
  Text(4, 0, 'Myntra.com'),
  Text(5, 0, 'Amazon.in, Flipkart.com'),
  Text(6, 0, 'Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com'),
  Text(7, 0, 'Amazon.in, Flipkart.com, Paytm.com, Snapdeal.com'),
  Text(8, 0, 'Flipkart.com, Myntra.com'),
  Text(9, 0, 'Amazon.in')])
```

Data Visualization for Product Variety

```
In [72]: #Count Plot for "Product_Variety" column
print(retention["Product_Variety"].value_counts())
plt.figure(figsize=(10,5))
sns.countplot("Product_Variety",data=retention)
plt.xticks(rotation=90)
```

Amazon.in, Flipkart.com	130
Amazon.in	43
Amazon.in, Myntra.com	20
Flipkart.com, Myntra.com	15
Myntra.com	15
Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com	14
Amazon.in, Flipkart.com, Paytm.com	13
Flipkart.com	12
Paytm.com	7

Name: Product_Variety, dtype: int64

```
Out[72]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8]),
 [Text(0, 0, 'Flipkart.com'),
  Text(1, 0, 'Flipkart.com, Myntra.com'),
  Text(2, 0, 'Amazon.in, Myntra.com'),
  Text(3, 0, 'Amazon.in, Flipkart.com'),
  Text(4, 0, 'Myntra.com'),
  Text(5, 0, 'Amazon.in, Flipkart.com, Paytm.com'),
  Text(6, 0, 'Amazon.in'),
  Text(7, 0, 'Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com'),
  Text(8, 0, 'Paytm.com')])
```

Data Visualization for Complete Product Information

```
In [73]: #Count Plot for "Complete_Product_Information" column
print(retention["Complete_Product_Information"].value_counts())
plt.figure(figsize=(10,5))
sns.countplot("Complete_Product_Information",data=retention)
plt.xticks(rotation=90)
```

Amazon.in, Flipkart.com	100
Amazon.in	43
Amazon.in, Flipkart.com, Paytm.com	24
Amazon.in, Paytm.com, Myntra.com	20
Amazon.in, Flipkart.com, Myntra.com	15
Amazon.in, Flipkart.com, Paytm.com, Myntra.com, Snapdeal.com	15
Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com	14
Snapdeal.com	12
Flipkart.com, Snapdeal.com	11
Flipkart.com	8
Amazon.in, Flipkart.com, Snapdeal.com	7
Name: Complete_Product_Information, dtype: int64	

```
Out[73]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10]),
 [Text(0, 0, 'Snapdeal.com'),
  Text(1, 0, 'Amazon.in, Flipkart.com, Myntra.com'),
  Text(2, 0, 'Amazon.in, Paytm.com, Myntra.com'),
  Text(3, 0, 'Amazon.in, Flipkart.com'),
  Text(4, 0, 'Amazon.in, Flipkart.com, Paytm.com, Myntra.com, Snapdeal.com'),
  Text(5, 0, 'Amazon.in, Flipkart.com, Paytm.com'),
  Text(6, 0, 'Flipkart.com, Snapdeal.com'),
  Text(7, 0, 'Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com'),
  Text(8, 0, 'Amazon.in'),
  Text(9, 0, 'Amazon.in, Flipkart.com, Snapdeal.com'),
  Text(10, 0, 'Flipkart.com')])
```

Data Visualization for Fast Loading Speed Web App

```
In [74]: #Count Plot for "Fast>Loading_Speed_Web_App" column
print(retention[ "Fast>Loading_Speed_Web_App" ].value_counts())
plt.figure(figsize=(10,5))
sns.countplot("Fast>Loading_Speed_Web_App",data=retention)
plt.xticks(rotation=90)
```

Amazon.in	51
Amazon.in, Paytm.com	44
Amazon.in, Flipkart.com, Myntra.com	30
Amazon.in, Flipkart.com, Paytm.com, Myntra.com, Snapdeal.com	30
Amazon.in, Flipkart.com	30
Amazon.in, Flipkart.com, Snapdeal.com	25
Amazon.in, Flipkart.com, Paytm.com	25
Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com	14
Snapdeal.com	12
Flipkart.com	8

Name: Fast>Loading_Speed_Web_App, dtype: int64

```
Out[74]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
 [Text(0, 0, 'Snapdeal.com'),
  Text(1, 0, 'Amazon.in, Flipkart.com, Myntra.com'),
  Text(2, 0, 'Amazon.in, Paytm.com'),
  Text(3, 0, 'Amazon.in, Flipkart.com, Snapdeal.com'),
  Text(4, 0, 'Amazon.in'),
  Text(5, 0, 'Amazon.in, Flipkart.com, Paytm.com, Myntra.com, Snapdeal.com'),
  Text(6, 0, 'Amazon.in, Flipkart.com, Paytm.com'),
  Text(7, 0, 'Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com'),
  Text(8, 0, 'Amazon.in, Flipkart.com'),
  Text(9, 0, 'Flipkart.com')])
```

Data Visualization for Reliable Website Application

```
In [75]: #Count Plot for "Reliable_Website_Application" column
print(retention["Reliable_Website_Application"].value_counts())
plt.figure(figsize=(10,5))
sns.countplot("Reliable_Website_Application",data=retention)
plt.xticks(rotation=90)
```

Amazon.in	61
Amazon.in, Flipkart.com	50
Amazon.in, Flipkart.com, Paytm.com	36
Amazon.in, Paytm.com, Myntra.com	35
Amazon.in, Flipkart.com, Snapdeal.com	18
Myntra.com	15
Flipkart.com	15
Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com	14
Amazon.in, Flipkart.com, Paytm.com, Snapdeal.com	13
Paytm.com	12
Name: Reliable_Website_Application, dtype: int64	

```
Out[75]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
[Text(0, 0, 'Paytm.com'),
 Text(1, 0, 'Myntra.com'),
 Text(2, 0, 'Amazon.in, Paytm.com, Myntra.com'),
 Text(3, 0, 'Amazon.in, Flipkart.com, Paytm.com'),
 Text(4, 0, 'Amazon.in, Flipkart.com, Paytm.com, Snapdeal.com'),
 Text(5, 0, 'Amazon.in, Flipkart.com, Snapdeal.com'),
 Text(6, 0, 'Amazon.in'),
 Text(7, 0, 'Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com'),
 Text(8, 0, 'Amazon.in, Flipkart.com'),
 Text(9, 0, 'Flipkart.com')])
```

Data Visualization for Quick Purchase

```
In [76]: #Count Plot for "Quick_Purchase" column
print(retention["Quick_Purchase"].value_counts())
plt.figure(figsize=(10,5))
sns.countplot("Quick_Purchase",data=retention)
plt.xticks(rotation=90)
```

Amazon.com	66
Amazon.com, Flipkart.com, Paytm.com	47
Amazon.com, Flipkart.com	37
Amazon.com, Flipkart.com, Myntra.com	30
Paytm.com	25
Amazon.com, Paytm.com, Myntra.com	20
Amazon.com, Flipkart.com, Paytm.com, Myntra.com, Snapdeal	15
Flipkart.com	15
Flipkart.com, Myntra.com, Snapdeal	14
Name: Quick_Purchase, dtype: int64	

```
Out[76]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8]),
 [Text(0, 0, 'Paytm.com'),
  Text(1, 0, 'Amazon.com, Flipkart.com, Myntra.com'),
  Text(2, 0, 'Amazon.com, Paytm.com, Myntra.com'),
  Text(3, 0, 'Amazon.com, Flipkart.com, Paytm.com'),
  Text(4, 0, 'Amazon.com, Flipkart.com, Paytm.com, Myntra.com, Snapdeal'),
  Text(5, 0, 'Amazon.com'),
  Text(6, 0, 'Flipkart.com, Myntra.com, Snapdeal'),
  Text(7, 0, 'Amazon.com, Flipkart.com'),
  Text(8, 0, 'Flipkart.com')])
```

Data Visualization for Payment Options

```
In [77]: #Count Plot for "Payment_Options" column  
print(retention["Payment_Options"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Payment_Options",data=retention)  
plt.xticks(rotation=90)
```

Amazon.in, Flipkart.com	65
Amazon.in, Flipkart.com, Myntra.com	40
Amazon.in, Flipkart.com, Patym.com, Myntra.com, Snapdeal.com	39
Amazon.in	23
Patym.com, Myntra.com	20
Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com	19
Amazon.in, Flipkart.com, Snapdeal.com	18
Flipkart.com, Myntra.com, Snapdeal.com	14
Patym.com	12
Amazon.in, Patym.com	11
Flipkart.com	8

Name: Payment_Options, dtype: int64

```
Out[77]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10]),  
 [Text(0, 0, 'Patym.com'),  
  Text(1, 0, 'Amazon.in, Flipkart.com, Myntra.com'),  
  Text(2, 0, 'Patym.com, Myntra.com'),  
  Text(3, 0, 'Amazon.in, Flipkart.com, Patym.com, Myntra.com, Snapdeal.com'),  
  Text(4, 0, 'Amazon.in, Flipkart.com, Snapdeal.com'),  
  Text(5, 0, 'Amazon.in, Flipkart.com'),  
  Text(6, 0, 'Flipkart.com, Myntra.com, Snapdeal.com'),  
  Text(7, 0, 'Amazon.in, Patym.com'),  
  Text(8, 0, 'Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com'),  
  Text(9, 0, 'Amazon.in'),  
  Text(10, 0, 'Flipkart.com')])
```

Data Visualization for Fast Delivery

```
In [78]: #Count Plot for "Fast_Delivery" column
print(retention["Fast_Delivery"].value_counts())
plt.figure(figsize=(10,5))
sns.countplot("Fast_Delivery",data=retention)
plt.xticks(rotation=90)
```

```
Amazon.in                               107
Amazon.in, Flipkart.com                  82
Amazon.in, Flipkart.com, Snapdeal.com   36
Amazon.in, Flipkart.com, Myntra.com     15
Flipkart.com                            15
Flipkart.com, Myntra.com, Snapdeal.com  14
Name: Fast_Delivery, dtype: int64
```

```
Out[78]: (array([0, 1, 2, 3, 4, 5]),
 [Text(0, 0, 'Amazon.in'),
  Text(1, 0, 'Amazon.in, Flipkart.com'),
  Text(2, 0, 'Amazon.in, Flipkart.com, Snapdeal.com'),
  Text(3, 0, 'Flipkart.com, Myntra.com, Snapdeal.com'),
  Text(4, 0, 'Amazon.in, Flipkart.com, Myntra.com'),
  Text(5, 0, 'Flipkart.com')])
```

Data Visualization for Customer Info Privacy

```
In [79]: #Count Plot for "Customer_Info_Privacy" column
print(retention["Customer_Info_Privacy"].value_counts())
plt.figure(figsize=(10,5))
sns.countplot("Customer_Info_Privacy",data=retention)
plt.xticks(rotation=90)
```

Amazon.in	71
Amazon.in, Flipkart.com	54
Amazon.in, Flipkart.com, Myntra.com	25
Amazon.in, Flipkart.com, Paytm.com, Myntra.com, Snapdeal.com	24
Paytm.com	18
Myntra.com	15
Amazon.in, Paytm.com	15
Flipkart.com	15
Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com	14
Amazon.in, Flipkart.com, Paytm.com	11
Amazon.in, Flipkart.com, Snapdeal.com	7
Name: Customer_Info_Privacy, dtype: int64	

```
out[79]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10]),  
 [Text(0, 0, 'Amazon.in'),  
  Text(1, 0, 'Myntra.com'),  
  Text(2, 0, 'Amazon.in, Flipkart.com, Myntra.com'),  
  Text(3, 0, 'Amazon.in, Paytm.com'),  
  Text(4, 0, 'Amazon.in, Flipkart.com, Paytm.com, Myntra.com, Snapdeal.com'),  
  Text(5, 0, 'Paytm.com'),  
  Text(6, 0, 'Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com'),  
  Text(7, 0, 'Amazon.in, Flipkart.com'),  
  Text(8, 0, 'Amazon.in, Flipkart.com, Paytm.com'),  
  Text(9, 0, 'Amazon.in, Flipkart.com, Snapdeal.com'),  
  Text(10, 0, 'Flipkart.com')])
```

Data Visualization for Customer Financial Info Security

```
In [80]: #Count Plot for "Customer_Financial_Info_Security" column
print(retention["Customer_Financial_Info_Security"].value_counts())
plt.figure(figsize=(10,5))
sns.countplot("Customer_Financial_Info_Security",data=retention)
plt.xticks(rotation=90)
```

Amazon.in	51
Amazon.in, Flipkart.com, Paytm.com, Myntra.com, Snapdeal.com	42
Flipkart.com	33
Amazon.in, Flipkart.com, Snapdeal.com	25
Amazon.in, Flipkart.com	24
Amazon.in, Paytm.com, Myntra.com	20
Amazon.in, Snapdeal.com	19
Myntra.com	15
Paytm.com	15
Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com	14
Amazon.in, Flipkart.com, Paytm.com	11
Name: Customer_Financial_Info_Security, dtype: int64	

```
Out[80]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10]),
 [Text(0, 0, 'Amazon.in'),
  Text(1, 0, 'Myntra.com'),
  Text(2, 0, 'Amazon.in, Paytm.com, Myntra.com'),
  Text(3, 0, 'Amazon.in, Flipkart.com, Snapdeal.com'),
  Text(4, 0, 'Paytm.com'),
  Text(5, 0, 'Amazon.in, Flipkart.com, Paytm.com, Myntra.com, Snapdeal.com'),
  Text(6, 0, 'Flipkart.com'),
  Text(7, 0, 'Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com'),
  Text(8, 0, 'Amazon.in, Flipkart.com'),
  Text(9, 0, 'Amazon.in, Flipkart.com, Paytm.com'),
  Text(10, 0, 'Amazon.in, Snapdeal.com')])
```

Data Visualization for Perceived Trustworthiness

```
In [81]: #Count Plot for "Perceived_Trustworthiness" column
print(retention["Perceived_Trustworthiness"].value_counts())
plt.figure(figsize=(10,5))
sns.countplot("Perceived_Trustworthiness",data=retention)
plt.xticks(rotation=90)
```

Amazon.in	76
Amazon.in, Flipkart.com, Snapdeal.com	36
Amazon.in, Myntra.com	35
Amazon.in, Flipkart.com	31
Flipkart.com	27
Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com	25
Myntra.com	15
Amazon.in, Flipkart.com, Paytm.com, Myntra.com, Snapdeal.com	13
Amazon.in, Flipkart.com, Paytm.com	11
Name: Perceived_Trustworthiness, dtype: int64	

```
Out[81]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8]),
[Text(0, 0, 'Flipkart.com'),
Text(1, 0, 'Myntra.com'),
Text(2, 0, 'Amazon.in, Myntra.com'),
Text(3, 0, 'Amazon.in, Flipkart.com, Snapdeal.com'),
Text(4, 0, 'Amazon.in, Flipkart.com, Paytm.com, Myntra.com, Snapdeal.com'),
Text(5, 0, 'Amazon.in'),
Text(6, 0, 'Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com'),
Text(7, 0, 'Amazon.in, Flipkart.com'),
Text(8, 0, 'Amazon.in, Flipkart.com, Paytm.com')])
```

Data Visualization for Multi Channel Assistance

```
In [82]: #Count Plot for "Multi_Channel_Assistance" column
print(retention["Multi_Channel_Assistance"].value_counts())
plt.figure(figsize=(10,5))
sns.countplot("Multi_Channel_Assistance",data=retention)
plt.xticks(rotation=90)
```

```
Amazon.in, Flipkart.com, Myntra.com, Snapdeal      61
Amazon.in                                         60
Amazon.in, Flipkart.com                         39
Amazon.in, Snapdeal                           26
Myntra.com                                       20
Amazon.in, Flipkart.com, Myntra.com            15
Amazon.in, Myntra.com                          15
Amazon.in, Flipkart.com, Paytm.com             13
Paytm.com                                         12
Flipkart.com                                     8
Name: Multi_Channel_Assistance, dtype: int64
```

```
Out[82]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
 [Text(0, 0, 'Paytm.com'),
  Text(1, 0, 'Amazon.in, Flipkart.com, Myntra.com'),
  Text(2, 0, 'Myntra.com'),
  Text(3, 0, 'Amazon.in, Flipkart.com, Myntra.com, Snapdeal'),
  Text(4, 0, 'Amazon.in, Myntra.com'),
  Text(5, 0, 'Amazon.in, Flipkart.com, Paytm.com'),
  Text(6, 0, 'Amazon.in'),
  Text(7, 0, 'Amazon.in, Flipkart.com'),
  Text(8, 0, 'Amazon.in, Snapdeal'),
  Text(9, 0, 'Flipkart.com')])
```

Data Visualization for Loading logging Time

```
In [83]: #Count Plot for "Loading_Logging_Time" column
print(retention["Loading_Logging_Time"].value_counts())
plt.figure(figsize=(10,5))
sns.countplot("Loading_Logging_Time",data=retention)
plt.xticks(rotation=90)
```

Amazon.in	57
Amazon.in, Flipkart.com	38
Paytm.com	38
Myntra.com	35
Amazon.in, Flipkart.com, Snapdeal.com	29
Snapdeal.com	25
Flipkart.com, Paytm.com	15
Flipkart.com, Paytm.com, Snapdeal.com	13
Amazon.in, Paytm.com	11
Flipkart.com	8

Name: Loading_Logging_Time, dtype: int64

```
out[83]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
 [Text(0, 0, 'Amazon.in'),
  Text(1, 0, 'Amazon.in, Flipkart.com'),
  Text(2, 0, 'Myntra.com'),
  Text(3, 0, 'Snapdeal.com'),
  Text(4, 0, 'Flipkart.com, Paytm.com'),
  Text(5, 0, 'Flipkart.com, Paytm.com, Snapdeal.com'),
  Text(6, 0, 'Amazon.in, Flipkart.com, Snapdeal.com'),
  Text(7, 0, 'Paytm.com'),
  Text(8, 0, 'Amazon.in, Paytm.com'),
  Text(9, 0, 'Flipkart.com')])
```

Data Visualization for Loading Graphic Photo Display time

```
In [84]: #Count Plot for "Loading_Graphics_Photo_Display_Time" column
print(retention["Loading_Graphics_Photo_Display_Time"].value_counts())
plt.figure(figsize=(10,5))
sns.countplot("Loading_Graphics_Photo_Display_Time",data=retention)
plt.xticks(rotation=90)
```

Amazon.in, Flipkart.com	60
Amazon.in	39
Mynta.com	35
Snapdeal.com	34
Mynta.com, Snapdeal.com	25
Flipkart.com, Snapdeal.com	19
Paytm.com	15
Flipkart.com	15
Amazon.in, Mynta.com, Snapdeal.com	14
Amazon.in, Paytm.com	13
Name: Loading_Graphics_Photo_Display_Time, dtype: int64	

```
Out[84]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
[Text(0, 0, 'Amazon.in'),
Text(1, 0, 'Mynta.com'),
Text(2, 0, 'Mynta.com, Snapdeal.com'),
Text(3, 0, 'Paytm.com'),
Text(4, 0, 'Amazon.in, Paytm.com'),
Text(5, 0, 'Amazon.in, Flipkart.com'),
Text(6, 0, 'Amazon.in, Mynta.com, Snapdeal.com'),
Text(7, 0, 'Flipkart.com, Snapdeal.com'),
Text(8, 0, 'Snapdeal.com'),
Text(9, 0, 'Flipkart.com')])
```

Data Visualization for Late Price Declare

```
In [85]: #Count Plot for "Late_Price_Declare" column
print(retention["Late_Price_Declare"].value_counts())
plt.figure(figsize=(10,5))
sns.countplot("Late_Price_Declare",data=retention)
plt.xticks(rotation=90)
```

```
Myntra.com          75
Paytm.com           52
snapdeal.com        41
Flipkart.com        38
Amazon.in           38
Amazon.in, Paytm.com    13
Paytm.com, snapdeal.com   7
Amazon.in, Flipkart.com   5
Name: Late_Price_Declare, dtype: int64
```

```
Out[85]: (array([0, 1, 2, 3, 4, 5, 6, 7]),
 [Text(0, 0, 'Flipkart.com'),
  Text(1, 0, 'snapdeal.com'),
  Text(2, 0, 'Myntra.com'),
  Text(3, 0, 'Paytm.com'),
  Text(4, 0, 'Amazon.in, Paytm.com'),
  Text(5, 0, 'Amazon.in'),
  Text(6, 0, 'Paytm.com, snapdeal.com'),
  Text(7, 0, 'Amazon.in, Flipkart.com')])
```

Data Visualization for Page Loading Time

```
In [86]: #Count Plot for "Page>Loading_Time" column
print(retention["Page>Loading_Time"].value_counts())
plt.figure(figsize=(10,5))
sns.countplot("Page>Loading_Time",data=retention)
plt.xticks(rotation=90)
```

Myntra.com	61
Paytm.com	59
Flipkart.com	32
Snapdeal.com	23
Amazon.in, Flipkart.com	18
Amazon.in	16
Paytm.com, Snapdeal.com	15
Amazon.in, Snapdeal.com	14
Amazon.in, Paytm.com	13
Flipkart.com, Snapdeal.com	11
Amazon.in, Paytm.com, Myntra.com	7
Name: Page>Loading_Time, dtype: int64	

```
out[86]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10]),
           [Text(0, 0, 'Flipkart.com'),
            Text(1, 0, 'Snapdeal.com'),
            Text(2, 0, 'Myntra.com'),
            Text(3, 0, 'Paytm.com'),
            Text(4, 0, 'Amazon.in, Paytm.com'),
            Text(5, 0, 'Amazon.in, Flipkart.com'),
            Text(6, 0, 'Amazon.in, Snapdeal.com'),
            Text(7, 0, 'Flipkart.com, Snapdeal.com'),
            Text(8, 0, 'Paytm.com, Snapdeal.com'),
            Text(9, 0, 'Amazon.in'),
            Text(10, 0, 'Amazon.in, Paytm.com, Myntra.com')])
```

Data Visualization for Limited payment mode

```
In [87]: #Count Plot for "Limited_Payment_Mode" column
print(retention["Limited_Payment_Mode"].value_counts())
plt.figure(figsize=(10,5))
sns.countplot("Limited_Payment_Mode",data=retention)
plt.xticks(rotation=90)
```

```
Snapdeal.com           87
Amazon.in              62
Flipkart.com            31
Amazon.in, Flipkart.com 29
Paytm.com               25
Paytm.com, Snapdeal.com 15
Amazon.in, Paytm.com      13
Myntra.com, Snapdeal.com    7
Name: Limited_Payment_Mode, dtype: int64
```

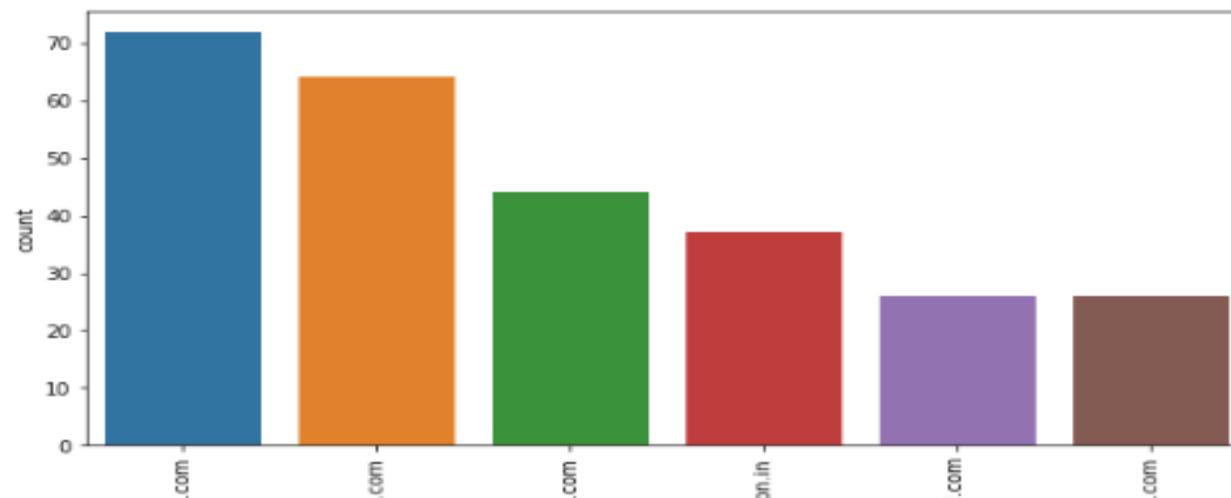
```
Out[87]: (array([0, 1, 2, 3, 4, 5, 6, 7]),
 [Text(0, 0, 'Amazon.in'),
  Text(1, 0, 'Snapdeal.com'),
  Text(2, 0, 'Paytm.com'),
  Text(3, 0, 'Amazon.in, Paytm.com'),
  Text(4, 0, 'Flipkart.com'),
  Text(5, 0, 'Amazon.in, Flipkart.com'),
  Text(6, 0, 'Paytm.com, Snapdeal.com'),
  Text(7, 0, 'Myntra.com, Snapdeal.com')])
```

Data Visualization for Late Delivery

```
In [88]: #Count Plot for "Late_Delivery" column
print(retention["Late_Delivery"].value_counts())
plt.figure(figsize=(10,5))
sns.countplot("Late_Delivery",data=retention)
plt.xticks(rotation=90)
```

```
Paytm.com           72
Snapdeal.com       64
Flipkart.com       44
Amazon.in          37
Paytm.com, Snapdeal.com 26
Myntra.com         26
Name: Late_Delivery, dtype: int64
```

```
out[88]: (array([0, 1, 2, 3, 4, 5]),
 [Text(0, 0, 'Paytm.com'),
  Text(1, 0, 'Snapdeal.com'),
  Text(2, 0, 'Flipkart.com'),
  Text(3, 0, 'Amazon.in'),
  Text(4, 0, 'Paytm.com, Snapdeal.com'),
  Text(5, 0, 'Myntra.com')])
```



Data Visualization for Web page performance

```
In [89]: #Count Plot for "Design_Change_of_Web_App" column  
print(retention["Design_Change_of_Web_App"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Design_Change_of_Web_App",data=retention)  
plt.xticks(rotation=90)
```

Amazon.in	96
Paytm.com	63
Amazon.in, Flipkart.com	45
Mynta.com	30
Flipkart.com	20
Snapdeal.com	8
Flipkart.com, Mynta.com	7
Name: Design_Change_of_Web_App, dtype: int64	

```
Out[89]: (array([0, 1, 2, 3, 4, 5, 6]),  
 [Text(0, 0, 'Flipkart.com'),  
  Text(1, 0, 'Amazon.in'),  
  Text(2, 0, 'Paytm.com'),  
  Text(3, 0, 'Amazon.in, Flipkart.com'),  
  Text(4, 0, 'Mynta.com'),  
  Text(5, 0, 'Snapdeal.com'),  
  Text(6, 0, 'Flipkart.com, Mynta.com')])
```

```
In [90]: #Count Plot for "Page_Disruption" column  
print(retention["Page_Disruption"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Page_Disruption",data=retention)  
plt.xticks(rotation=90)
```

Amazon.in	53
Mynta.com	52
Snapdeal.com	49
Paytm.com	39
Flipkart.com	26
Amazon.in, Flipkart.com	25
Mynta.com, Snapdeal.com	14
Flipkart.com, Snapdeal.com	11
Name: Page_Disruption, dtype: int64	

```
Out[90]: (array([0, 1, 2, 3, 4, 5, 6, 7]),  
 [Text(0, 0, 'Amazon.in'),  
  Text(1, 0, 'Mynta.com'),  
  Text(2, 0, 'Paytm.com'),  
  Text(3, 0, 'Amazon.in, Flipkart.com'),  
  Text(4, 0, 'Snapdeal.com'),  
  Text(5, 0, 'Flipkart.com'),  
  Text(6, 0, 'Mynta.com, Snapdeal.com'),  
  Text(7, 0, 'Flipkart.com, Snapdeal.com')])
```

Data Visualization for Website recommendation

```
In [91]: #Count Plot for "Website_Efficiency" column  
print(retention["Website_Efficiency"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Website_Efficiency",data=retention)  
plt.xticks(rotation=90)
```

Amazon.in	94
Flipkart.com	47
Amazon.in, Flipkart.com	45
Amazon.in, Flipkart.com, Paytm.com	25
Amazon.in, Paytm.com	18
Paytm.com	15
Myntra.com, Snapdeal.com	14
Snapdeal.com	11
Name: Website_Efficiency, dtype: int64	

```
out[91]: (array([0, 1, 2, 3, 4, 5, 6, 7]),  
 [Text(0, 0, 'Amazon.in'),  
  Text(1, 0, 'Amazon.in, Flipkart.com'),  
  Text(2, 0, 'Amazon.in, Flipkart.com, Paytm.com'),  
  Text(3, 0, 'Paytm.com'),  
  Text(4, 0, 'Snapdeal.com'),  
  Text(5, 0, 'Myntra.com, Snapdeal.com'),  
  Text(6, 0, 'Flipkart.com'),  
  Text(7, 0, 'Amazon.in, Paytm.com')])
```

```
In [92]: #Count Plot for "Recommend_to_Friend" column  
print(retention["Recommend_to_Friend"].value_counts())  
plt.figure(figsize=(10,5))  
sns.countplot("Recommend_to_Friend",data=retention)  
plt.xticks(rotation=90)
```

Amazon.in	79
Amazon.in, Flipkart.com	62
Flipkart.com	39
Amazon.in, Myntra.com	30
Amazon.in, Paytm.com, Myntra.com	20
Amazon.in, Flipkart.com, Myntra.com	15
Amazon.in, Paytm.com	13
Flipkart.com, Paytm.com, Myntra.com, snapdeal.com	11
Name: Recommend_to_Friend, dtype: int64	

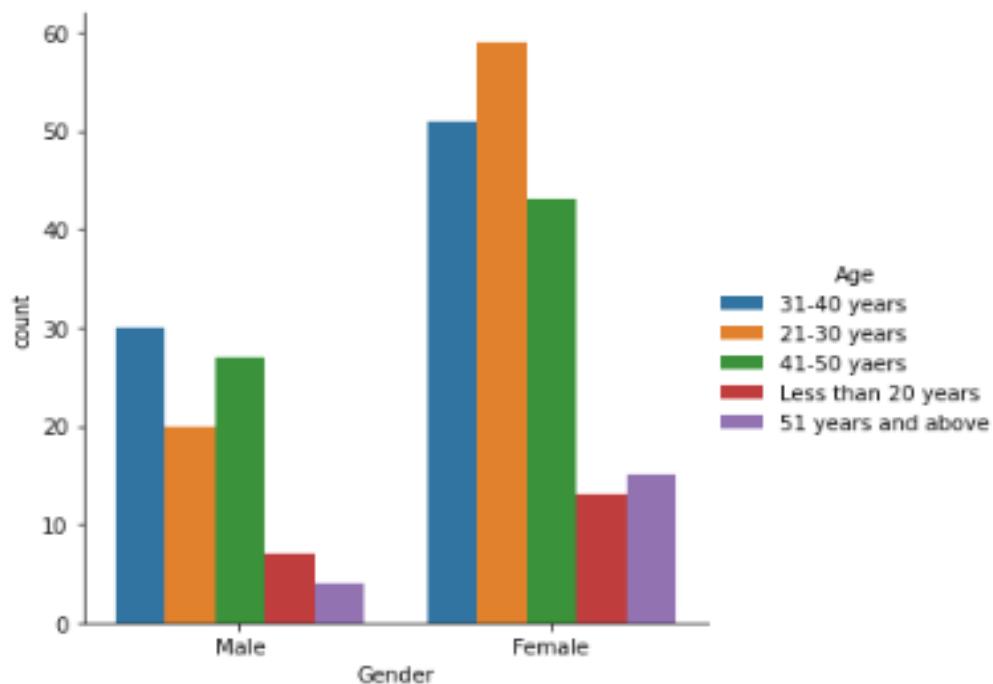
```
out[92]: (array([0, 1, 2, 3, 4, 5, 6, 7]),  
 [Text(0, 0, 'Flipkart.com'),  
  Text(1, 0, 'Amazon.in, Myntra.com'),  
  Text(2, 0, 'Amazon.in, Paytm.com, Myntra.com'),  
  Text(3, 0, 'Amazon.in, Flipkart.com'),  
  Text(4, 0, 'Amazon.in, Paytm.com'),  
  Text(5, 0, 'Flipkart.com, Paytm.com, Myntra.com, snapdeal.com'),  
  Text(6, 0, 'Amazon.in'),  
  Text(7, 0, 'Amazon.in, Flipkart.com, Myntra.com')])
```

Bivariate Analysis

```
In [93]: #factorplot for comparision between "Gender" and "Age" column  
plt.figure(figsize=(10,5))  
sns.factorplot("Gender", hue="Age", data=retention, kind='count')
```

```
Out[93]: <seaborn.axisgrid.FacetGrid at 0xa02ab69b20>
```

<Figure size 720x360 with 0 Axes>



We can see Male customers of:

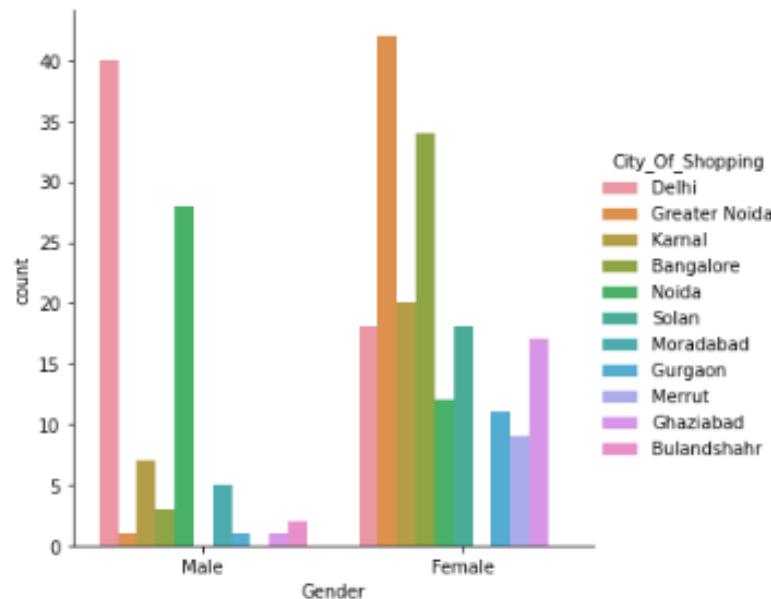
Age 31-40 years : 30
Age 21-30 years : 20
Age 41-50 years : 28
Age less than 20 years : 8
Age 51 years and above : 5

And Female Customers of:

Age 31-40 years : 50
Age 21-30 years : 58
Age 41-50 years : 42
Age less than 20 years : 12
Age 51 years and above : 15

Observation & Conclusion

```
In [94]: #factorplot for comparision between "Gender" and "City_Of_Shopping" column  
plt.figure(figsize=(10,5))  
sns.factorplot("Gender", hue="City_Of_Shopping", data=retention, kind='count')  
  
Out[94]: <seaborn.axisgrid.FacetGrid at 0xa02a9ea850>  
  
<Figure size 720x360 with 0 Axes>
```

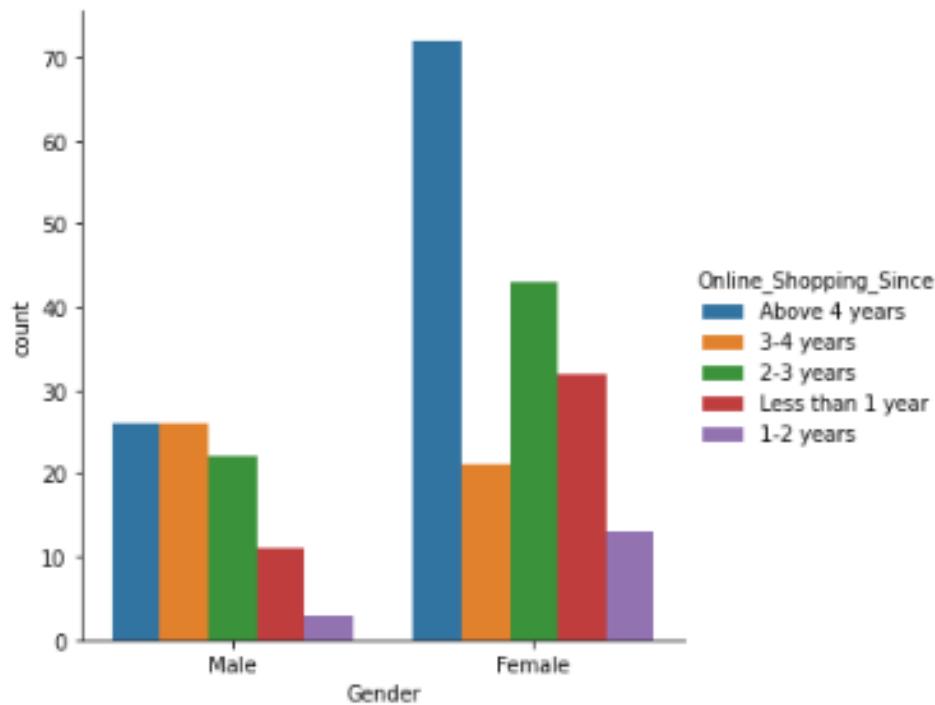


Female Customers from Greater Noida is more compare to other city and Male Customers from Delhi is more compare to other city. ALso Female customer are more compare to male customer.

Observation & Conclusion

```
In [95]: #factorplot for comparision between "Gender" and "Online_Shopping_Since" column  
plt.figure(figsize=(10,5))  
sns.factorplot("Gender", hue="Online_Shopping_Since", data=retention, kind='count')
```

```
Out[95]: <seaborn.axisgrid.FacetGrid at 0xa02ab93790>  
  
<Figure size 720x360 with 0 Axes>
```

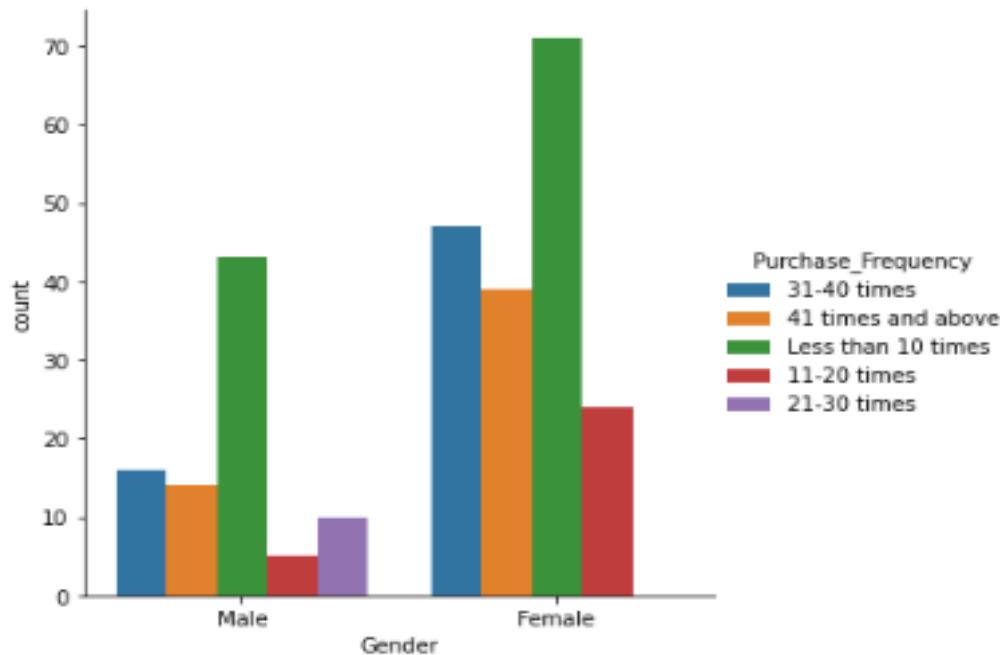


Female customer shopping from above 4years are more than 70.

Observation & Conclusion

```
In [96]: #factorplot for comparision between "Gender" and "Purchase_Frequency" column  
plt.figure(figsize=(10,5))  
sns.factorplot("Gender", hue="Purchase_Frequency", data=retention, kind='count')
```

```
out[96]: <seaborn.axisgrid.FacetGrid at 0xa02be35100>  
<Figure size 720x360 with 0 Axes>
```



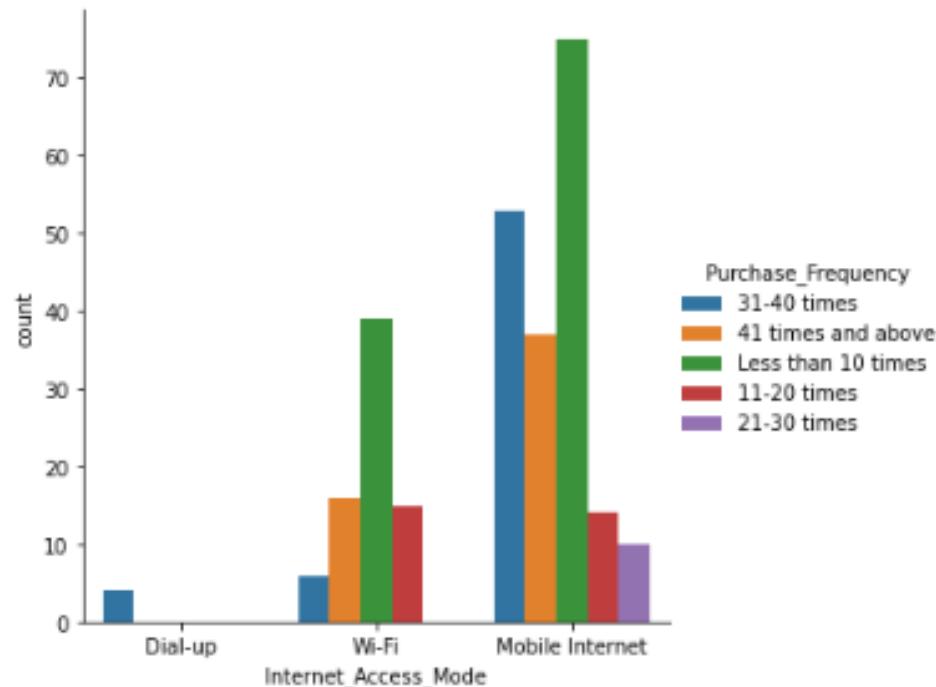
Female customer purchasing online Less than 10 times are 70 and male ecustomer are 44.

Observation & Conclusion

```
In [97]: #factorplot for comparision between "Internet_Access_Mode" and "Purchase_Frequency" column  
plt.figure(figsize=(10,5))  
sns.factorplot("Internet_Access_Mode", hue="Purchase_Frequency", data=retention, kind='count')
```

```
out[97]: <seaborn.axisgrid.FacetGrid at 0xa02be1a0d0>
```

```
<Figure size 720x360 with 0 Axes>
```

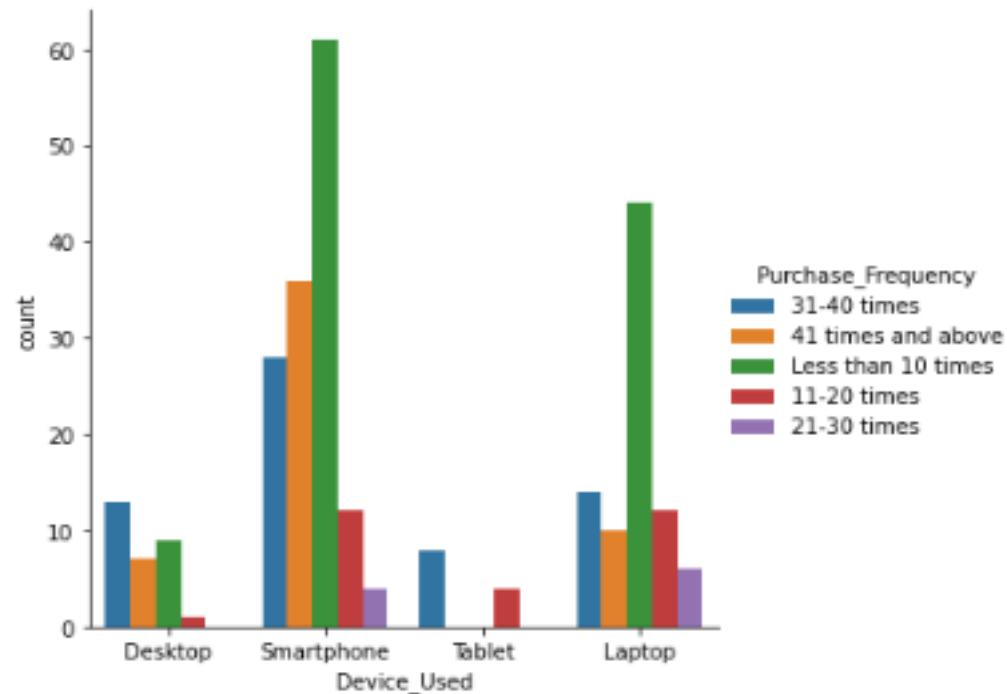


Through Mobile Internet Purchase was done frequently less than 10 times.

Observation & Conclusion

```
In [98]: #factorplot for comparision between "Device_Used" and "Purchase_Frequency" column  
plt.figure(figsize=(10,5))  
sns.factorplot("Device_Used", hue="Purchase_Frequency", data=retention, kind='count')
```

```
Out[98]: <seaborn.axisgrid.FacetGrid at 0xa02bee9be0>  
<Figure size 720x360 with 0 Axes>
```

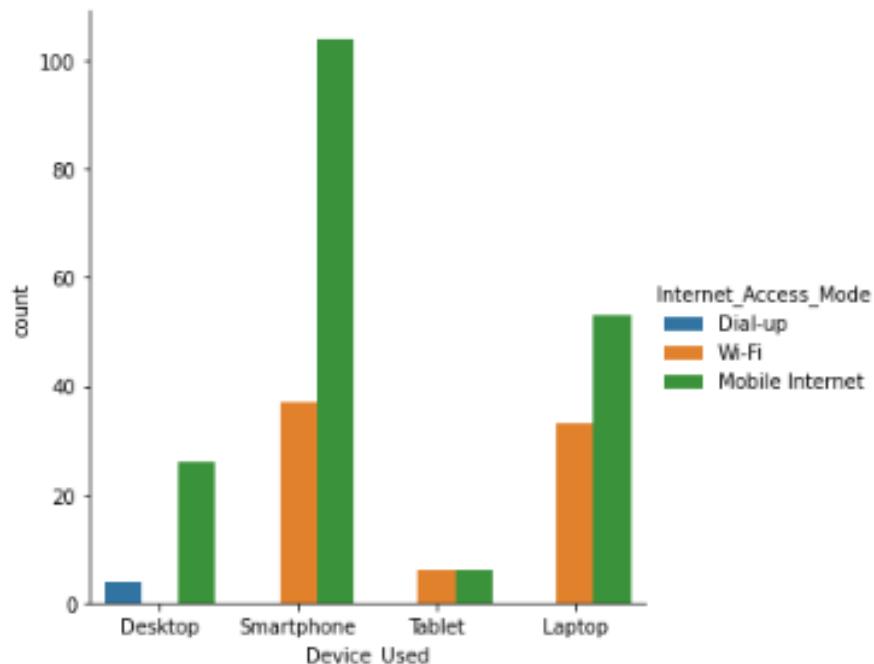


Smartphone is used to purchase frequently.

Observation & Conclusion

```
In [99]: #factorplot for comparision between "Device_Used" and "Internet_Access_Mode" column  
plt.figure(figsize=(10,5))  
sns.factorplot("Device_Used", hue="Internet_Access_Mode", data=retention, kind='count')
```

```
Out[99]: <seaborn.axisgrid.FacetGrid at 0xa02c03d370>  
<Figure size 720x360 with 0 Axes>
```



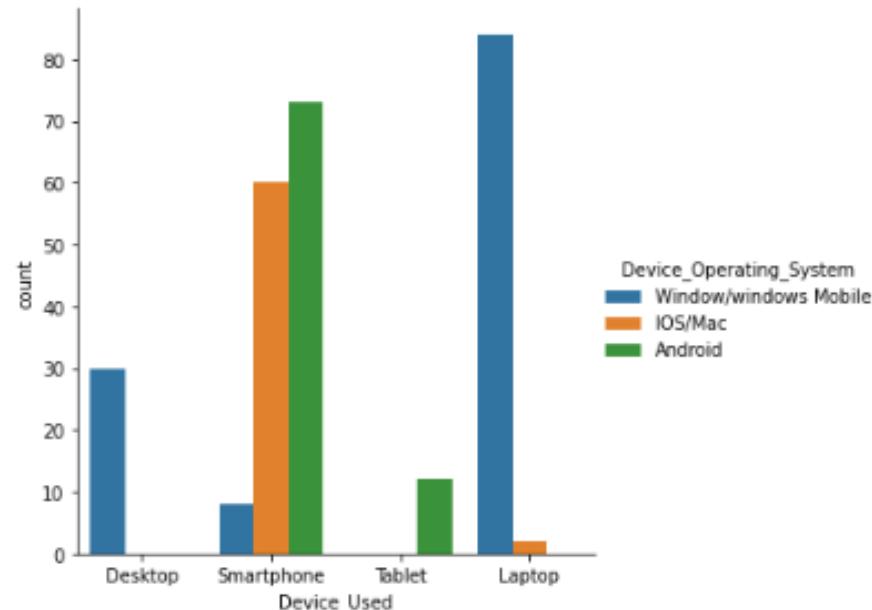
Mobile Internet is used mostly for Internet Access in Smartphone, Laptop, Desktop & Tablet and Dial-up is used least in Desktop only.

Observation & Conclusion

```
In [100]: #factorplot for comparision between "Device_Used" and "Device_Operating_System" column  
plt.figure(figsize=(10,5))  
sns.factorplot("Device_Used", hue="Device_Operating_System", data=retention, kind='count')
```

```
Out[100]: <seaborn.axisgrid.FacetGrid at 0xa02bf95ca0>
```

```
<Figure size 720x360 with 0 Axes>
```



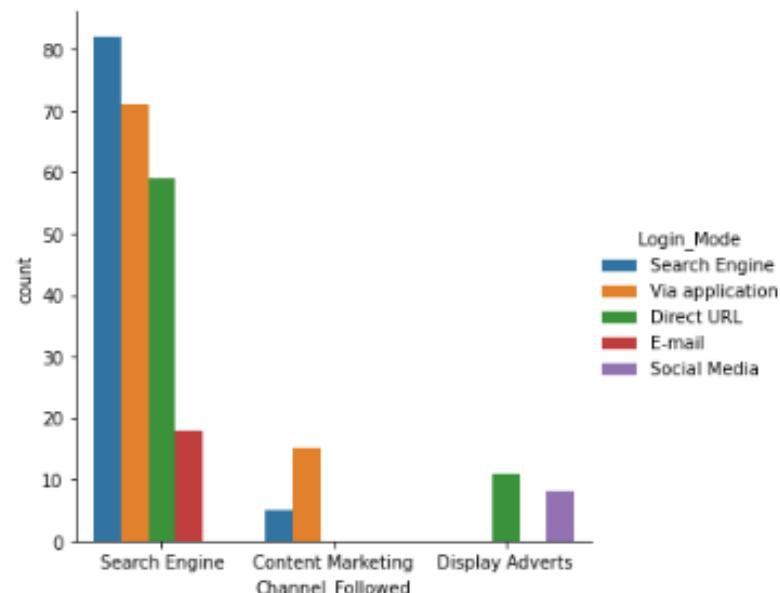
Operating system Windows/windows Mobile is used in Laptop Most then in Desktop and least in Smartphone. Android is used most in Smartphone then IOS/Mac O/S. In Tablet only Android O/s is used.

Observation & Conclusion

```
In [101]: #factorplot for comparision between "Channel_Followed" and "Login_Mode" column  
plt.figure(figsize=(20,5))  
sns.factorplot("Channel_Followed", hue="Login_Mode", data=retention, kind='count')
```

```
out[101]: <seaborn.axisgrid.FacetGrid at 0xa02c036400>
```

```
<Figure size 1440x360 with 0 Axes>
```

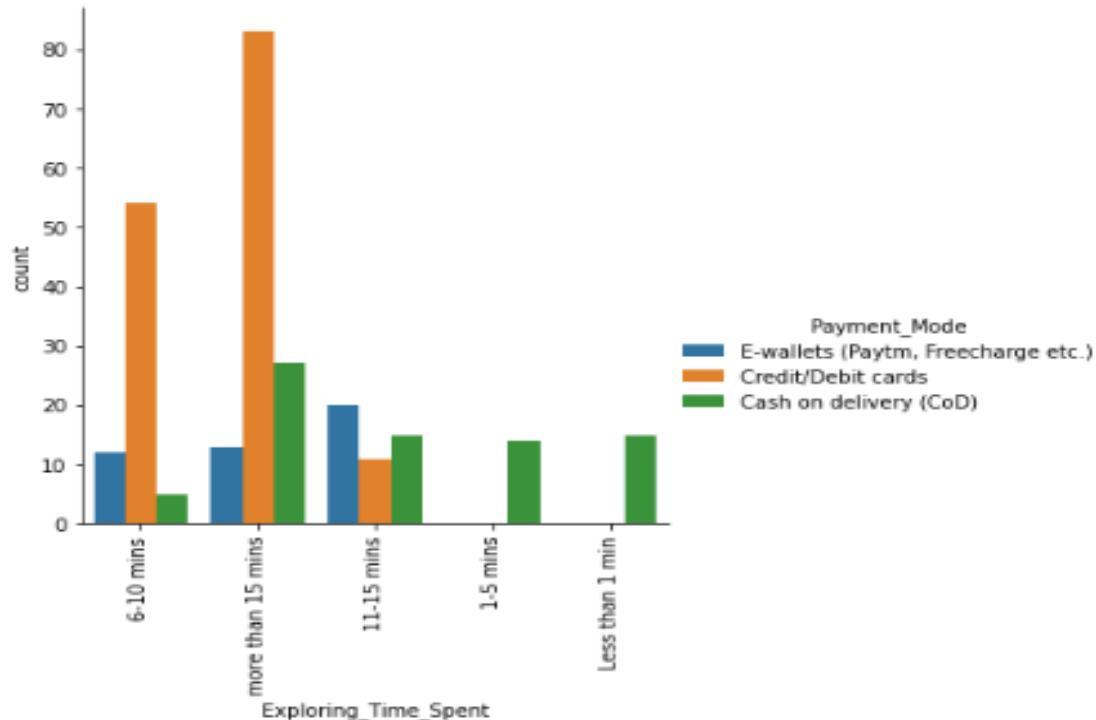


Most customers uses Search Engine to visit the online store for first time and after first visit they Login using Search Engine again. So, Search Engine is the best mode.

Observation & Conclusion

```
In [102]: #factorplot for comparision between "Exploring_Time_Spent" and "Payment_Mode" column  
sns.factorplot("Exploring_Time_Spent", hue="Payment_Mode", data=retention, kind='count')  
plt.xticks(rotation=90)
```

```
out[102]: (array([0, 1, 2, 3, 4]),  
 [Text(0, 0, '6-10 mins'),  
  Text(1, 0, 'more than 15 mins'),  
  Text(2, 0, '11-15 mins'),  
  Text(3, 0, '1-5 mins'),  
  Text(4, 0, 'Less than 1 min')])
```

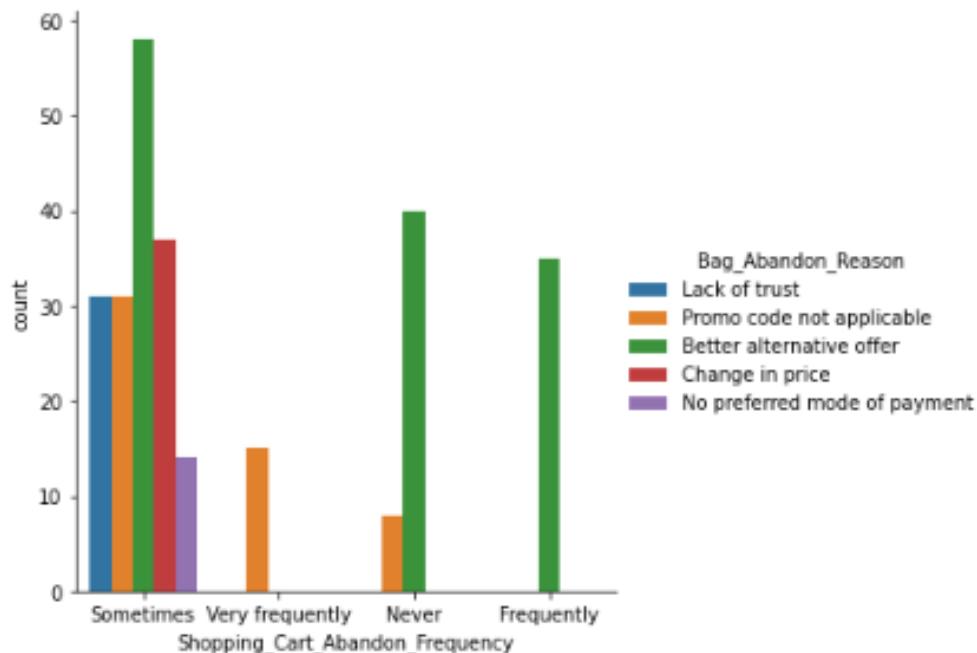


Most customers take more than 15 mins to make purchase decision and uses credit/debit card as payment option mostly.

Observation & Conclusion

```
In [103]: #factorplot for comparision between "Shopping_Cart_Abandon_Frequency" and "Bag_Abandon_Reason" column
plt.figure(figsize=(10,5))
sns.factorplot("Shopping_Cart_Abandon_Frequency", hue="Bag_Abandon_Reason", data=retention, kind='count')

Out[103]: <seaborn.axisgrid.FacetGrid at 0xa02d23c0a0>
<Figure size 720x360 with 0 Axes>
```



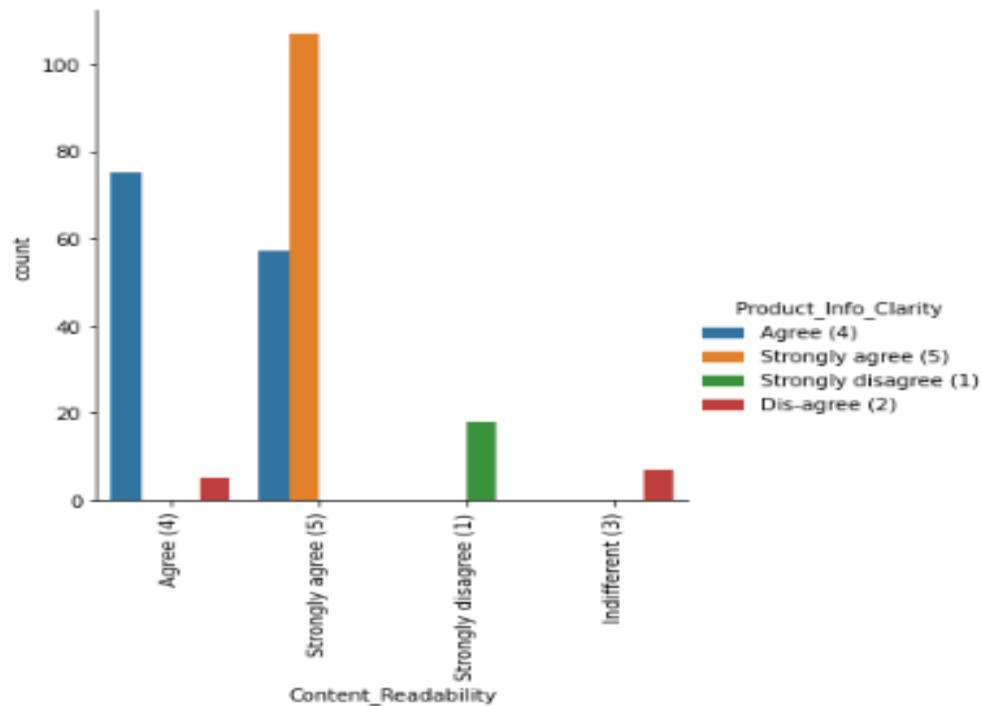
There are 5 reason to abandon Bag/Shopping cart but Top 2 reasons are:

- Sometimes and frequently Bag/Shopping cart is abandon due to better alternative offer mostly is first reason and*
- second reason is Sometimes due to change in price

Observation & Conclusion

```
In [104]: #factorplot for comparision between "Content_Readability" and "Product_Info_Clarity" column
plt.figure(figsize=(10,5))
sns.factorplot("Content_Readability", hue="Product_Info_Clarity", data=retention, kind='count')
plt.xticks(rotation=90)

out[104]: (array([0, 1, 2, 3]),
 [Text(0, 0, 'Agree (4)'),
  Text(1, 0, 'Strongly agree (5)'),
  Text(2, 0, 'Strongly disagree (1)'),
  Text(3, 0, 'Indifferent (3)')])
<Figure size 720x360 with 0 Axes>
```

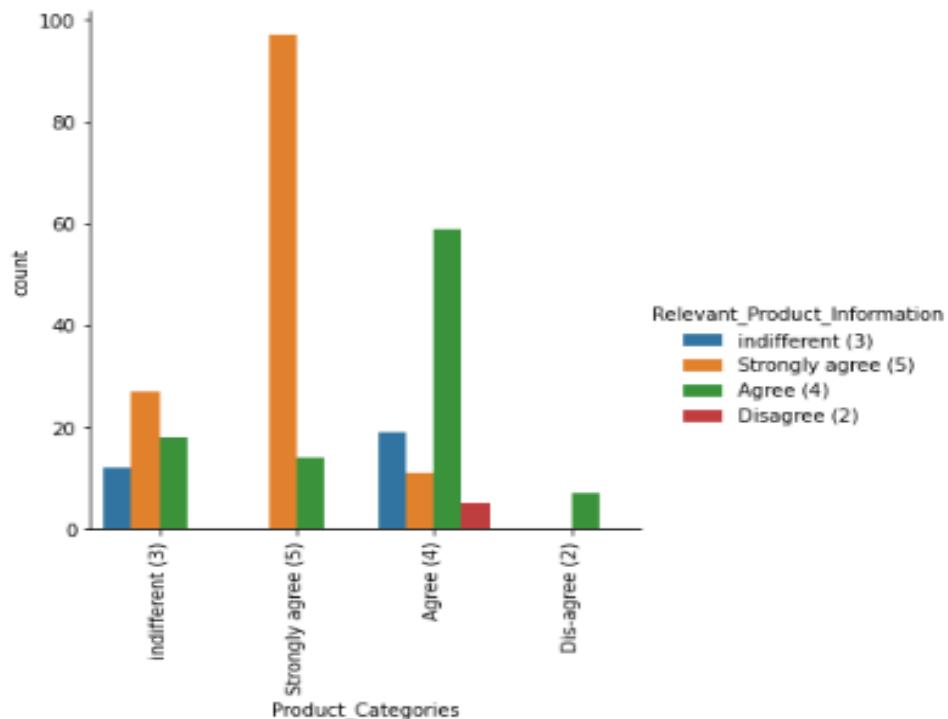


According to customer Content Readability and Product Info Clarity is must and Strongly Agree to it.

Observation & Conclusion

```
In [105]: #factorplot for comparision between "Product_Categories" and "Relevant_Product_Information" column
sns.factorplot("Product_Categories", hue="Relevant_Product_Information", data=retention, kind='count')
plt.xticks(rotation=90)
```

```
out[105]: (array([0, 1, 2, 3]),
 [Text(0, 0, 'indifferent (3)'),
 Text(1, 0, 'Strongly agree (5)'),
 Text(2, 0, 'Agree (4)'),
 Text(3, 0, 'Dis-agree (2)')])
```

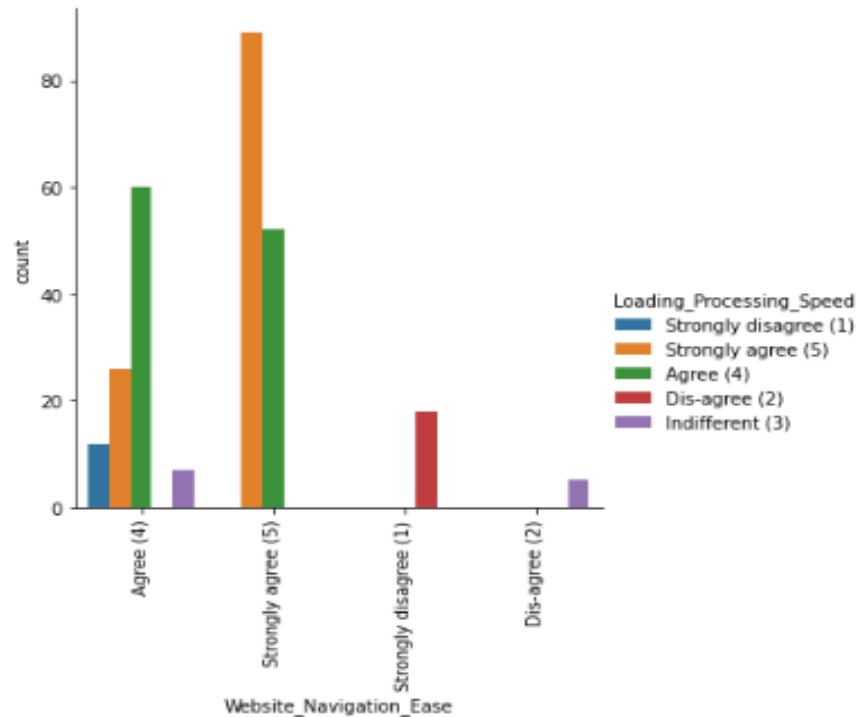


Strongly agree that Relevant Product Information and Product Categories are must according to customer rating.

Observation & Conclusion

```
In [106]: #factorplot for comparision between "Website_Navigation_Ease" and "Loading_Processing_Speed" column
sns.factorplot("Website_Navigation_Ease", hue="Loading_Processing_Speed", data=retention, kind='count')
plt.xticks(rotation=90)
```

```
Out[106]: (array([0, 1, 2, 3]),
 [Text(0, 0, 'Agree (4)'),
 Text(1, 0, 'Strongly agree (5)'),
 Text(2, 0, 'Strongly disagree (1)'),
 Text(3, 0, 'Dis-agree (2)')])
```

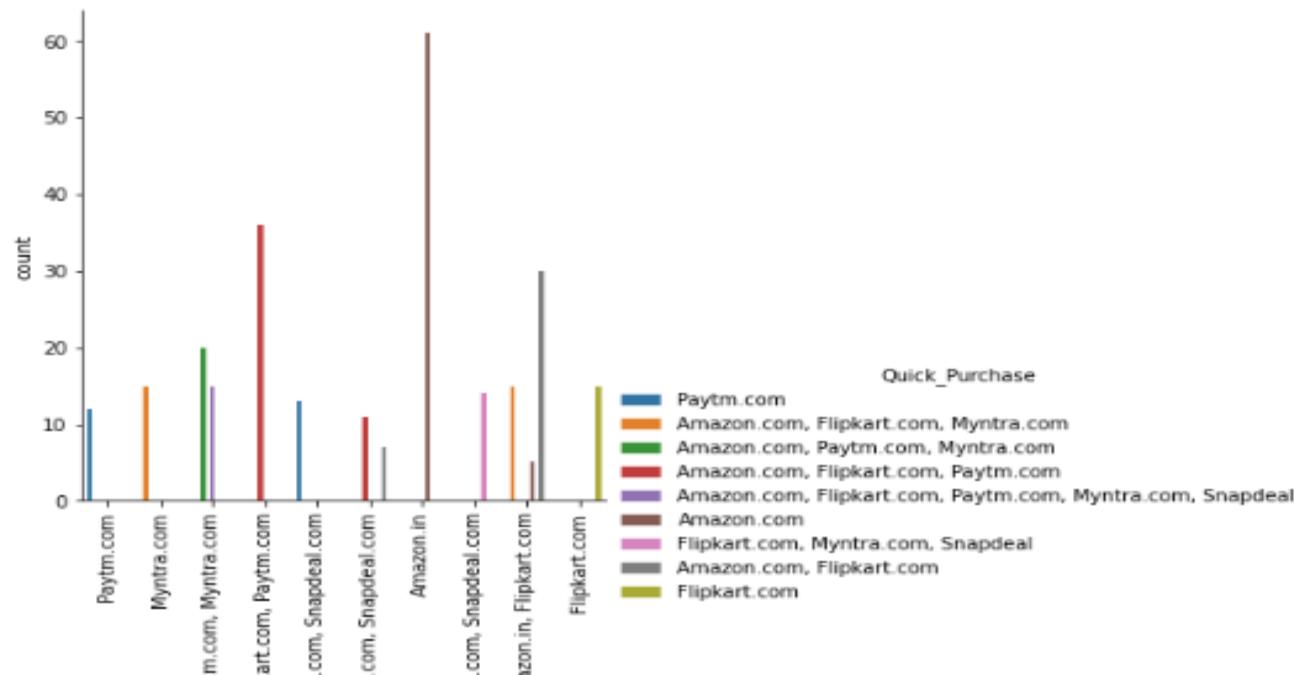


Most customers agrees to get easy navigation in website and they strongly agree to have loading and processing speed fast of website.

Observation & Conclusion

```
In [107]: #factorplot for comparision between "Reliable_Website_Application" and "Quick_Purchase" column  
sns.factorplot("Reliable_Website_Application", hue="Quick_Purchase", data=retention, kind='count')  
plt.xticks(rotation=90)
```

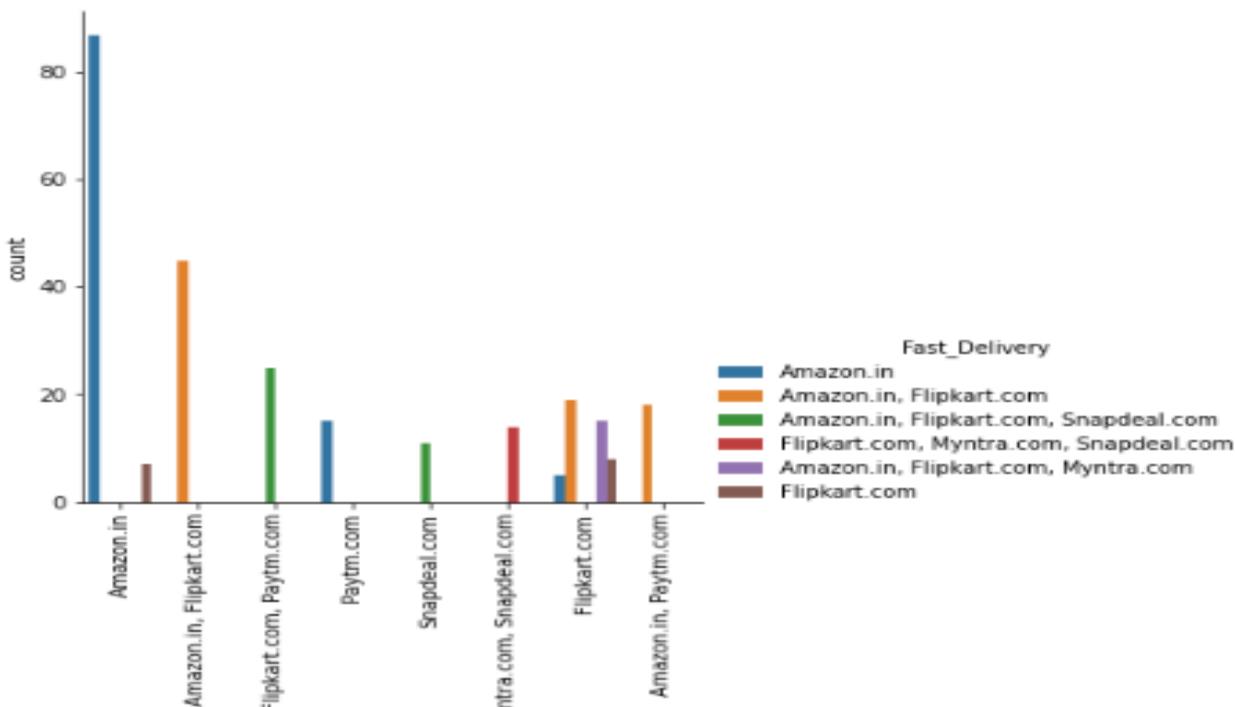
```
Out[107]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),  
 [Text(0, 0, 'Paytm.com'),  
  Text(1, 0, 'Myntra.com'),  
  Text(2, 0, 'Amazon.in, Paytm.com, Myntra.com'),  
  Text(3, 0, 'Amazon.in, Flipkart.com, Paytm.com'),  
  Text(4, 0, 'Amazon.in, Flipkart.com, Paytm.com, Snapdeal.com'),  
  Text(5, 0, 'Amazon.in, Flipkart.com, Snapdeal.com'),  
  Text(6, 0, 'Amazon.in'),  
  Text(7, 0, 'Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com'),  
  Text(8, 0, 'Amazon.in, Flipkart.com'),  
  Text(9, 0, 'Flipkart.com')])
```



Observation & Conclusion

```
In [108]: #factorplot for comparision between "Website_Efficiency" and "Fast_Delivery" column  
sns.factorplot("Website_Efficiency", hue="Fast_Delivery", data=retention, kind='count')  
plt.xticks(rotation=90)
```

```
out[108]: (array([0, 1, 2, 3, 4, 5, 6, 7]),  
 [Text(0, 0, 'Amazon.in'),  
  Text(1, 0, 'Amazon.in, Flipkart.com'),  
  Text(2, 0, 'Amazon.in, Flipkart.com, Paytm.com'),  
  Text(3, 0, 'Paytm.com'),  
  Text(4, 0, 'Snapdeal.com'),  
  Text(5, 0, 'Myntra.com, Snapdeal.com'),  
  Text(6, 0, 'Flipkart.com'),  
  Text(7, 0, 'Amazon.in, Paytm.com')])
```



Observation & Conclusion

- ❖ Efficient Website with Fastest Delivery is Amazon. In according to customers.
- ❖ High Customer Satisfaction from website:
 - i. Amazon.com
 - ii. Flipkart.com
- ❖ High Risk from website:
 - i. Myntra.com
 - ii. Snapdeal.com
- ❖ Website recommendation to a friend:
 - i. Amazon. In
 - ii. Flipkart.com

Observation & Conclusion

- ❖ Most Customer believe that the content on the website must be easy to read and understand.
- ❖ Most Customer agree that information on similar product to the one highlighted is important for product comparison.
- ❖ Most Customer (70%) believe that Complete information of listed seller and product is important for purchase decision.
- ❖ Most Customer (90%) agree Shopping online is convenient and flexible.
- ❖ Most Customer (70%) believe that Enjoyment is derived from shopping online.
- ❖ Most Customer believe that Online shopping gives monetary benefit and discounts.
- ❖ Getting value for money spent on Online Shopping.
- ❖ Return and replacement policy is important for purchase decision according to 90% customers.
- ❖ Gaining access to loyalty programs is a benefit of shopping online.
- ❖ Most Customer (90%) believe that All relevant information on listed products must be stated clearly.
- ❖ Most Customer (80%) believe that Displaying quality Information on the website improves satisfaction of customers.

Observation & Conclusion

- ❖ **More than 90% Customer wants website to be:**
 - a. Easy to navigate
 - b. Loading and processing speed fast
 - c. User friendly Interface
 - d. Convenient Payment methods
 - e. Guarantee the privacy of the customer
 - f. Availability of several communication channels (email, online rep, twitter, phone etc.)

Observation & Conclusion

- ❖ **60-80 % Customer believe that:**
 - a. Shopping on your preferred Retailer enhances your social status
 - b. You feel gratification shopping on your favorite retailer
 - c. Shopping on the website helps you fulfill certain roles

Observation & Conclusion

❖ Customer "Strongly Agree" (rating 5) to:

- i. Shopping on a good quality website or application
- ii. Net Benefit derived from shopping online
- iii. Customer Trust
- iv. Offering a wide variety of listed product in several category
- v. Provision of complete and relevant product information
- vi. Monetary savings
- vii. The Convenience of patronizing the online retailer
- viii. Shopping on the website gives you the sense of adventure

Closing Note

E-commerce to take into account their customer satisfaction because this will retain customer loyalty as well as attract potential customers

Thank
you

