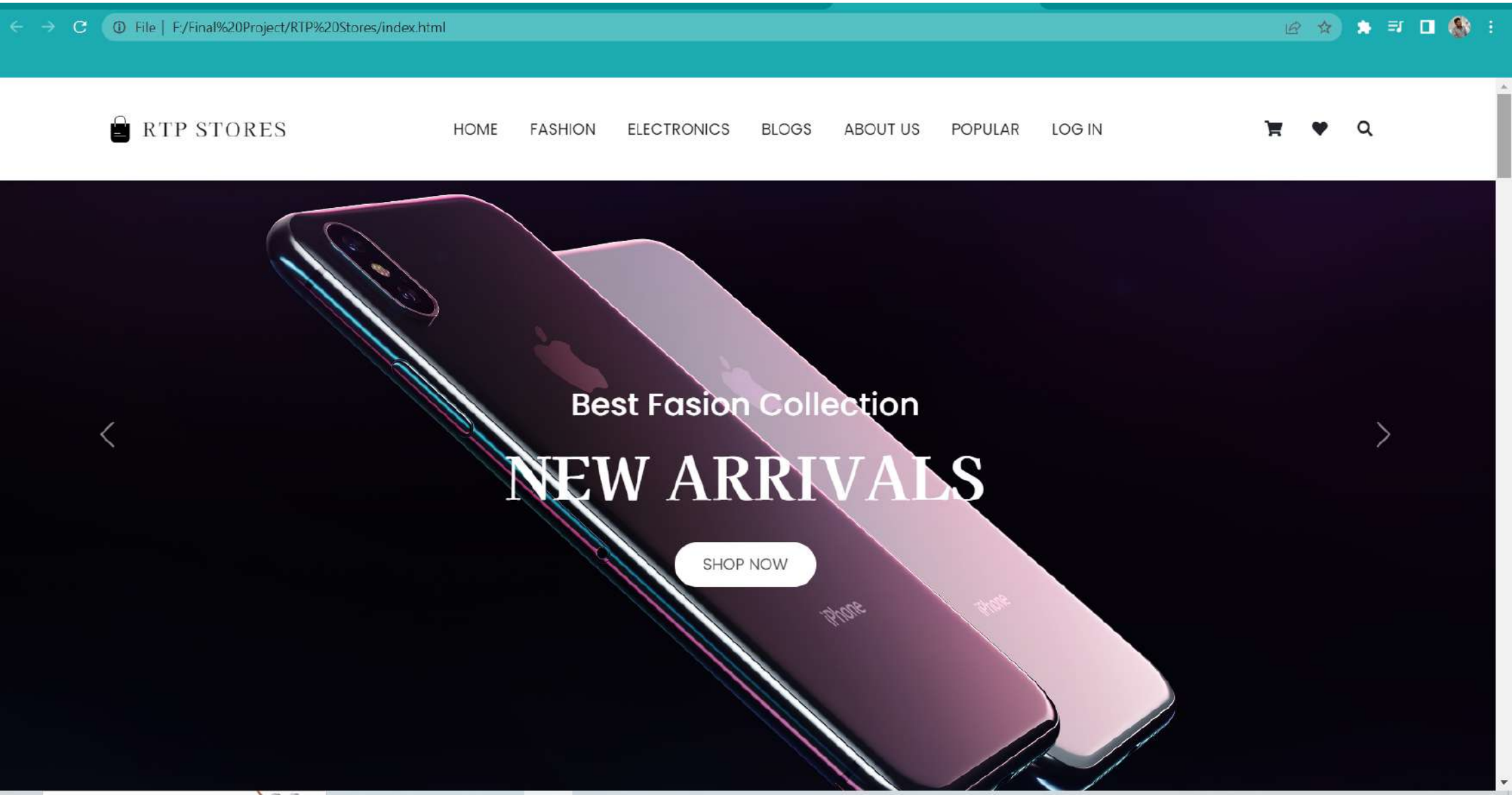

File | F:/Final%20Project/RTP%20Stores/index.html
RTP STORES
HOME
FASHION
ELECTRONICS
BLOGS
ABOUT US
POPULAR
LOG IN
Best Fasion Collection
NEW ARRIVALS
SHOP NOW

File | F:/Final%20Project/RTP%20Stores/index.html

RTP STORES

HOME FASHION ELECTRONICS BLOGS ABOUT US POPULAR LOG IN

New Collection

All | Best Sellers | Featured | New Arrival

★★★★★
Apple iPhone 12 64 GB (Red, 4 GB RAM)
₹ 50,900

★★★★★
OnePlus Nord CE 2 Lite 5G (Blue Tide, 6GB RAM, 128GB Storage)
₹ 18,900

★★★★★
HP Victus Gaming Latest 12th Gen Intel Core i7 FHD Gaming Laptop (8GB RAM/512GB SSD/RTX 3060 4GB Graphics)
₹ 82,990

★★★★★
Dell New Windows Inspiron 7425 2in1 Laptop, AMD Ryzen5-5625U 8 GB GDDR4, 512GB SSD,
₹ 77,800

★★★★★
HP Pavilion 14 12th Gen Intel Core i7 16GB SDRAM/1TB SSD 14 Inch(35.6cm) FHD
₹ 84,490

★★★★★
Apple 2020 MacBook Air Laptop M1 Chip, 13.3-Inch 8GB RAM, 256GB SSD Storage
₹ 84490

★★★★★
Realme Narzo 50 Pro 5G (Hyper Black 6GB RAM+128GB Storage)| 50% Charge In 31 Min
₹ 17,249

★★★★★
iQOO Neo 6 5G (Dark Nova, 8GB RAM, 128GB Storage) | Only Snapdragon 870 in The Segment | 50% Charge in Just 12 Mins
₹ 24,990

File | F:/Final%20Project/RTP%20Stores/index.html

RTP STORES

HOME FASHION ELECTRONICS BLOGS ABOUT US POPULAR LOG IN

Special Selection

Apple iPhone 14 256 GB (Blue, 6 GB RAM)(P>
₹ 85,900
Add To Cart

Apple 2022 MacBook Air Laptop With M2 Chip: 13.6-Inch, 8GB RAM, 256GB SSD Storage
₹ 1,06,990
Add to Cart

Samsung Galaxy S23 5G (Green, 8GB, 256GB Storage)
₹ 79,999
Add to Cart

Apple 2020 MacBook Air Laptop M1 Chip, 13.3-Inch 8GB RAM, 256GB SSD Storage
₹ 84490
Add to Cart

File | F:/Final%20Project/RTP%20Stores/index.html

RTP STORES

HOME FASHION ELECTRONICS BLOGS ABOUT US POPULAR LOG IN

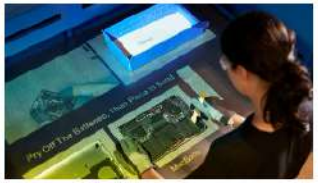
Our Latest Blog

SwiftKey keyboard for Android and iOS get Bing AI

Microsoft unveiled its latest update for SwiftKey, making the Bing AI experience easily accessible with just a single touch on any iOS or Android device that supports third-party keyboards. This comes after Microsoft recently rolled out Bing Chat AI into SwiftKey beta for Android.

Author: Tamil G

Read More



Apple to use 100% recycled cobalt in batteries by 2025

Apple has announced a new goal to speed up its use of recycled materials in its products, with a 2025 deadline to make all Apple-designed batteries from 100 percent recycled cobalt.This is part of Apple's increased efforts to reuse gold, tungsten, cobalt, and other elements in its products, which it started last year.

Author: Tamil G

Read More

Apple triples India iPhone output to $7 Billion: Report

Apple has tripled its iPhone production in India to over $7 billion in the last fiscal year, as it seeks to diversify its supply chain and tap into the world's fastest-growing smartphone market – according to a recent report by Bloomberg.The company has also announced that it will launch its first retail stores in India – Apple BKC in Mumbai and Apple Saket in Delhi

Author: Tamil G

Read More

About Us

We are the leading e-commerce company that specializes in providing high-quality products to customers worldwide.

Our mission is to offer a seamless shopping experience through our user-friendly website and mobile app, with a wide range of products to meet the diverse needs of our customers.

Read More

File | F:/Final%20Project/RTP%20Stores/index.html

RTP STORES

HOME FASHION ELECTRONICS BLOGS ABOUT US POPULAR LOG IN

Read More

Popular Of This Year

Top Rated

Vivo Y22 (Starlit Blue, 6GB RAM 128GB Storage)
₹ 16,499

Redmi 11 Prime 5G (Meadow Green, 6GB RAM, 128GB Storage)
₹ 15,599

Vivo Y16 (Drizzling Gold, 4GB RAM 64GB Storage)
₹ 12,499

Best Selling

realme Narzo 50 Pro 5G (Hyper Blue 8GB RAM+128GB Storage)
₹ 18,749

Samsung Galaxy M33 5G (Mystique Green, 8GB, 128GB Storage)
₹ 19,999

Oppo A78 5G (Glowing Black, 8GB RAM, 128 Storage)
₹ 18,999

On Sale

Dell Vostro 3420 Laptop,12th Gen Intel Core i3-1215U, 8GB & 512GB SSD
₹ 41,990

HP Pavilion 14, AMD Ryzen 5-5625U 8GB RAM/512GB SSD 14inches
₹ 52,990

Lenovo IdeaPad Slim 3 Intel Core i5 11th Gen 15.6 inches 8GB/512GB SSD
₹ 50,990

Sign Up For Subscription

Explore More And Get Special Offers

Enter Your Email ... Log In

RTP STORES

Links
- Home
- Collection
- Blogs
- About Us

Contact Us
- Albert Street, Mumbai, India
- rtpstore.support@gmail.com
- +9188 7777 3333

Follow Us

Cohort Analysis

Quarter ..	Custome..	2018 Q1	2018 Q2	2018 Q3	2018 Q4	2019 Q1	2019 Q2	2019 Q3	2019 Q4	2020 Q1	2020 Q2	2020 Q3	2020 Q4	2021 Q1	2021 Q2	2021 Q3	2021 Q4
2018 Q1	63	100.0%	17.5%	17.5%	15.9%	7.9%	9.5%	14.3%	28.6%	6.3%	23.8%	19.0%	27.0%	11.1%	28.6%	31.7%	33.3%
2018 Q2	107		100.0%	11.2%	22.4%	8.4%	15.9%	15.9%	26.2%	13.1%	24.3%	20.6%	29.0%	16.8%	19.6%	23.4%	36.4%
2018 Q3	120			100.0%	18.3%	11.7%	15.8%	18.3%	23.3%	10.0%	20.0%	24.2%	30.8%	11.7%	25.8%	26.7%	30.0%
2018 Q4	132				100.0%	11.4%	12.9%	18.2%	19.7%	12.1%	15.2%	25.8%	25.8%	12.1%	25.0%	25.0%	41.7%
2019 Q1	32					100.0%	12.5%	21.9%	21.9%	9.4%	18.8%	18.8%	31.3%	15.6%	28.1%	25.0%	43.8%
2019 Q2	37						100.0%	16.2%	27.0%	13.5%	24.3%	16.2%	35.1%	13.5%	18.9%	21.6%	43.2%
2019 Q3	54							100.0%	24.1%	14.8%	13.0%	24.1%	16.7%	18.5%	27.8%	31.5%	42.6%
2019 Q4	62								100.0%	11.3%	19.4%	25.8%	21.0%	19.4%	19.4%	35.5%	38.7%
2020 Q1	22									100.0%	31.8%	22.7%	31.8%	4.5%	27.3%	22.7%	40.9%
2020 Q2	32										100.0%	28.1%	18.8%	12.5%	12.5%	34.4%	40.6%
2020 Q3	28											100.0%	17.9%	28.6%	25.0%	39.3%	25.0%
2020 Q4	27												100.0%	18.5%	18.5%	14.8%	51.9%
2021 Q1	11													100.0%	27.3%	36.4%	36.4%
2021 Q2	14														100.0%	21.4%	35.7%
2021 Q3	10															100.0%	40.0%
2021 Q4	17																100.0%

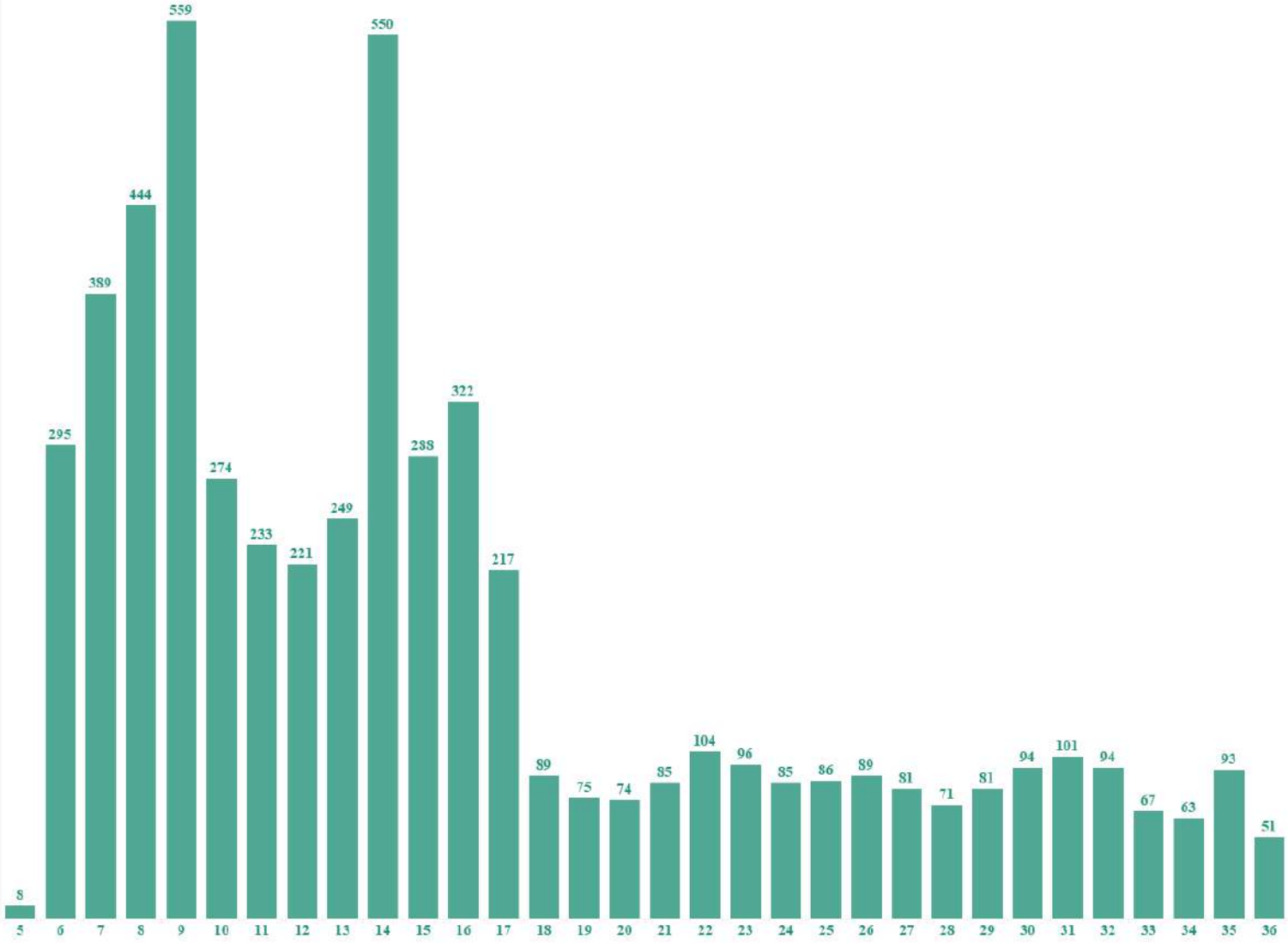
Activate Windows
Go to Settings to activate Windows.

Cohort Analysis

Quarter ..	Custome..	2018 Q1	2018 Q2	2018 Q3	2018 Q4	2019 Q1	2019 Q2	2019 Q3	2019 Q4	2020 Q1	2020 Q2	2020 Q3	2020 Q4	2021 Q1	2021 Q2	2021 Q3	2021 Q4
2018 Q1	63	100.0%	37.5%	43.8%	31.2%	12.5%	25.0%	56.2%	68.8%	18.8%	93.8%	62.5%	75.0%	6.2%	81.2%	50.0%	87.5%
2018 Q2	107		100.0%	14.3%	64.3%	0.0%	42.9%	57.1%	78.6%	7.1%	71.4%	28.6%	92.9%	50.0%	35.7%	21.4%	85.7%
2018 Q3	120			100.0%	23.1%	15.4%	7.7%	30.8%	69.2%	38.5%	46.2%	53.8%	84.6%	0.0%	61.5%	76.9%	92.3%
2018 Q4	132				100.0%	0.0%	8.3%	58.3%	50.0%	25.0%	41.7%	66.7%	83.3%	16.7%	33.3%	75.0%	91.7%
2019 Q1	32					100.0%	45.5%	81.8%	54.5%	0.0%	9.1%	27.3%	36.4%	18.2%	63.6%	72.7%	90.9%
2019 Q2	37						100.0%	40.0%	50.0%	20.0%	30.0%	10.0%	90.0%	0.0%	70.0%	60.0%	80.0%
2019 Q3	54							100.0%	33.3%	0.0%	11.1%	55.6%	44.4%	22.2%	77.8%	66.7%	88.9%
2019 Q4	62								100.0%	0.0%	75.0%	50.0%	12.5%	37.5%	25.0%	62.5%	87.5%
2020 Q1	22									100.0%	71.4%	42.9%	57.1%	0.0%	28.6%	14.3%	85.7%
2020 Q2	32										100.0%	66.7%	33.3%	0.0%	16.7%	83.3%	50.0%
2020 Q3	28											100.0%	40.0%	60.0%	0.0%	80.0%	20.0%
2020 Q4	27												100.0%	50.0%	0.0%	25.0%	75.0%
2021 Q1	11													100.0%	66.7%	0.0%	33.3%
2021 Q2	14														100.0%	0.0%	50.0%
2021 Q3	10															100.0%	0.0%
2021 Q4	17																100.0%

Activate Windows
Go to Settings to activate Windows.

Products Delivered Vs Distance Graph



Activate Windows
Go to Settings to activate Windows.

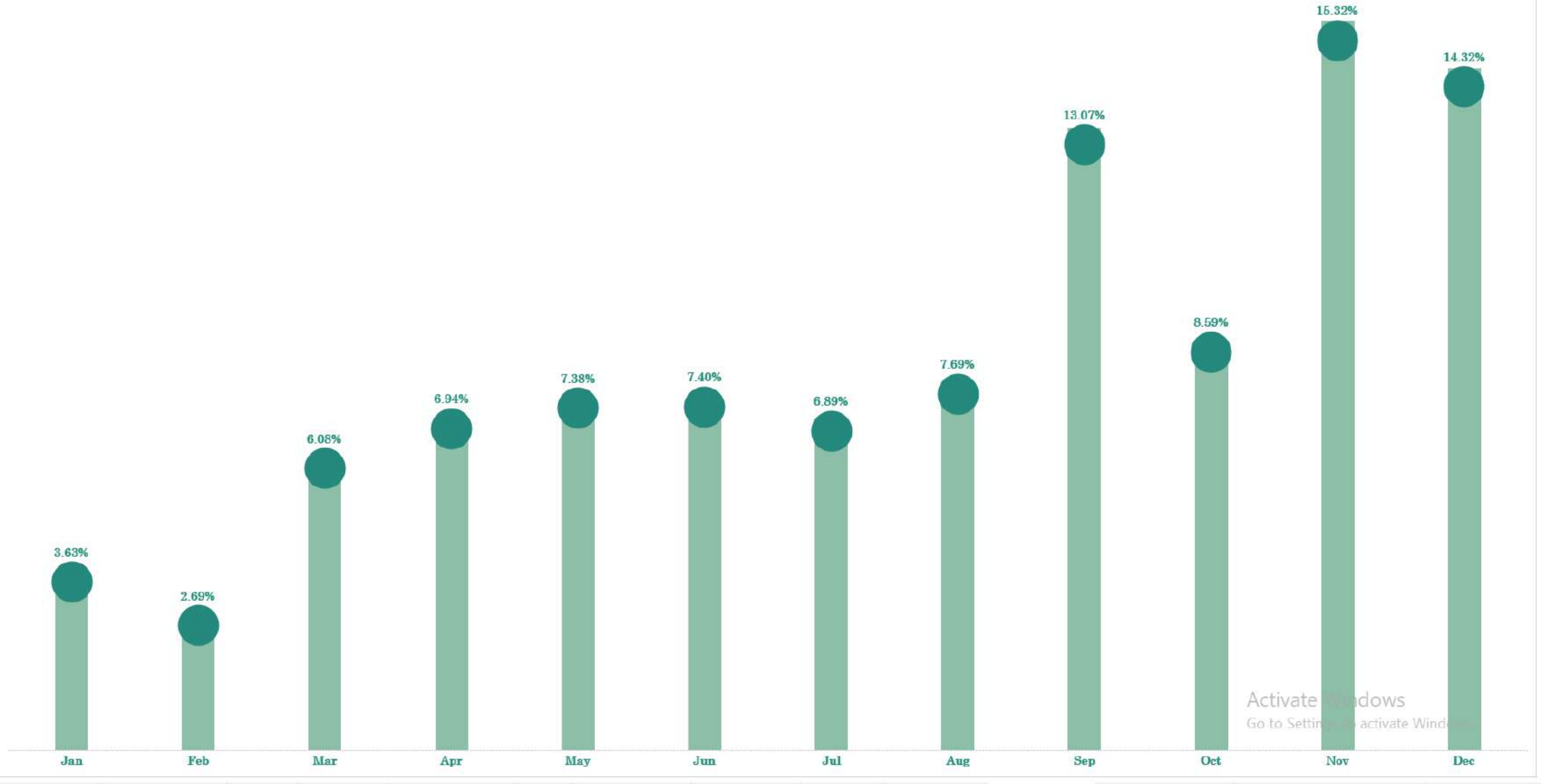
Gender Based Customer Analysis

Activate Windows
Go to Settings to activate Windows.

Products solds as per the City-tier

Activate Windows
Go to Settings to activate Windows.

Profit Per Month

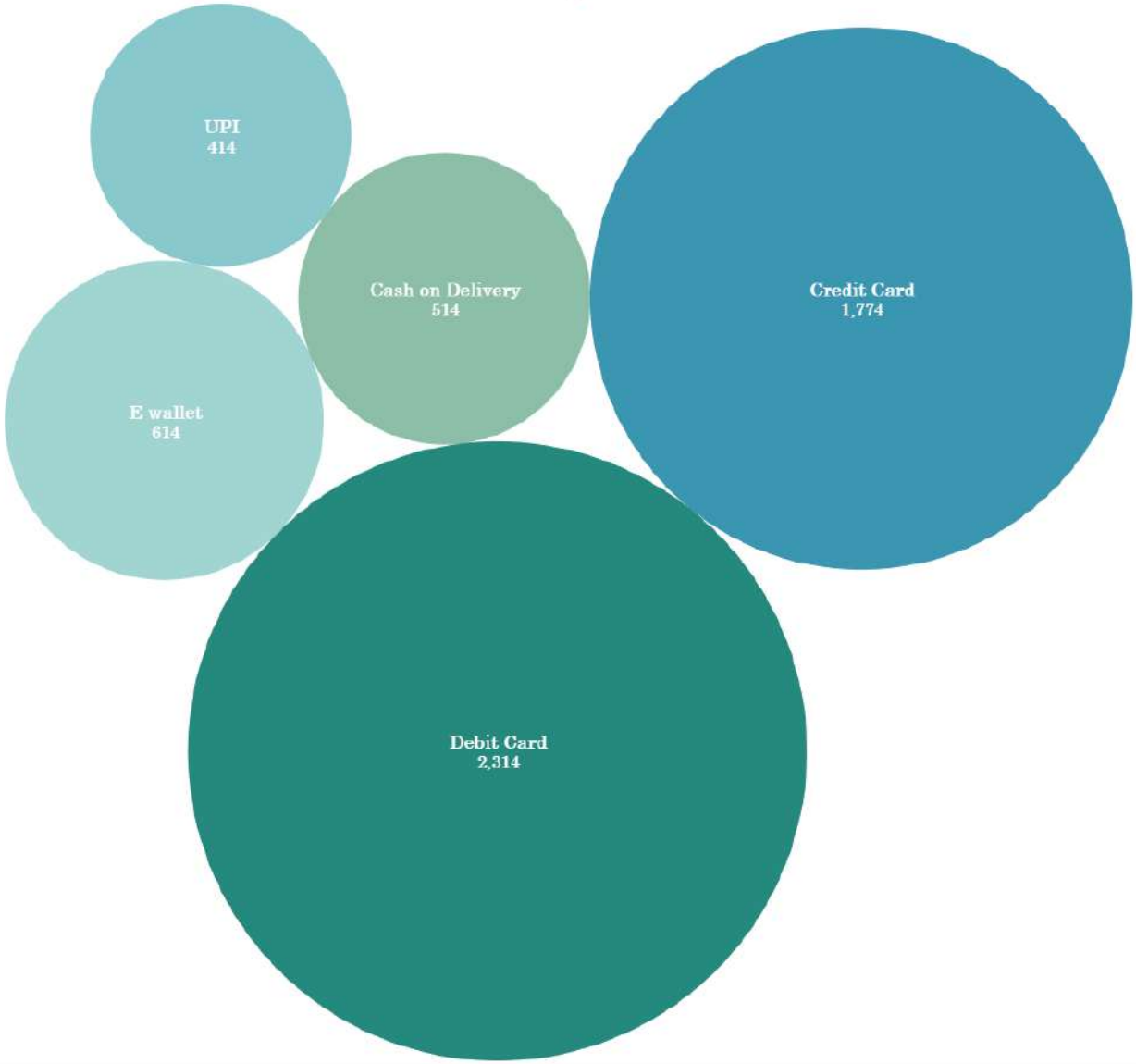


Activate Windows
Go to Settings to activate Windows.

Champion Products

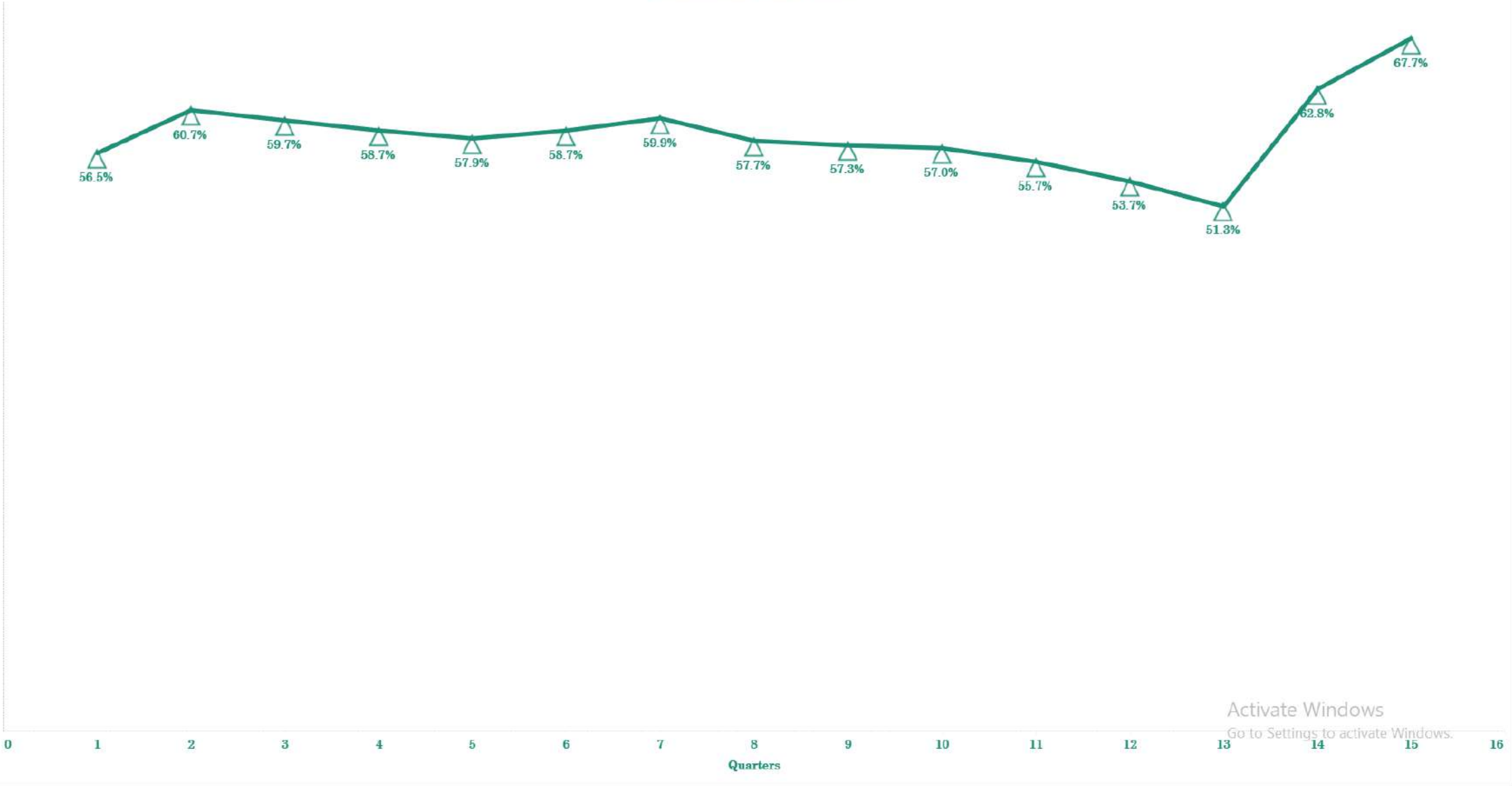
Activate Windows
Go to Settings to activate Windows.

Preferred Payment Mode

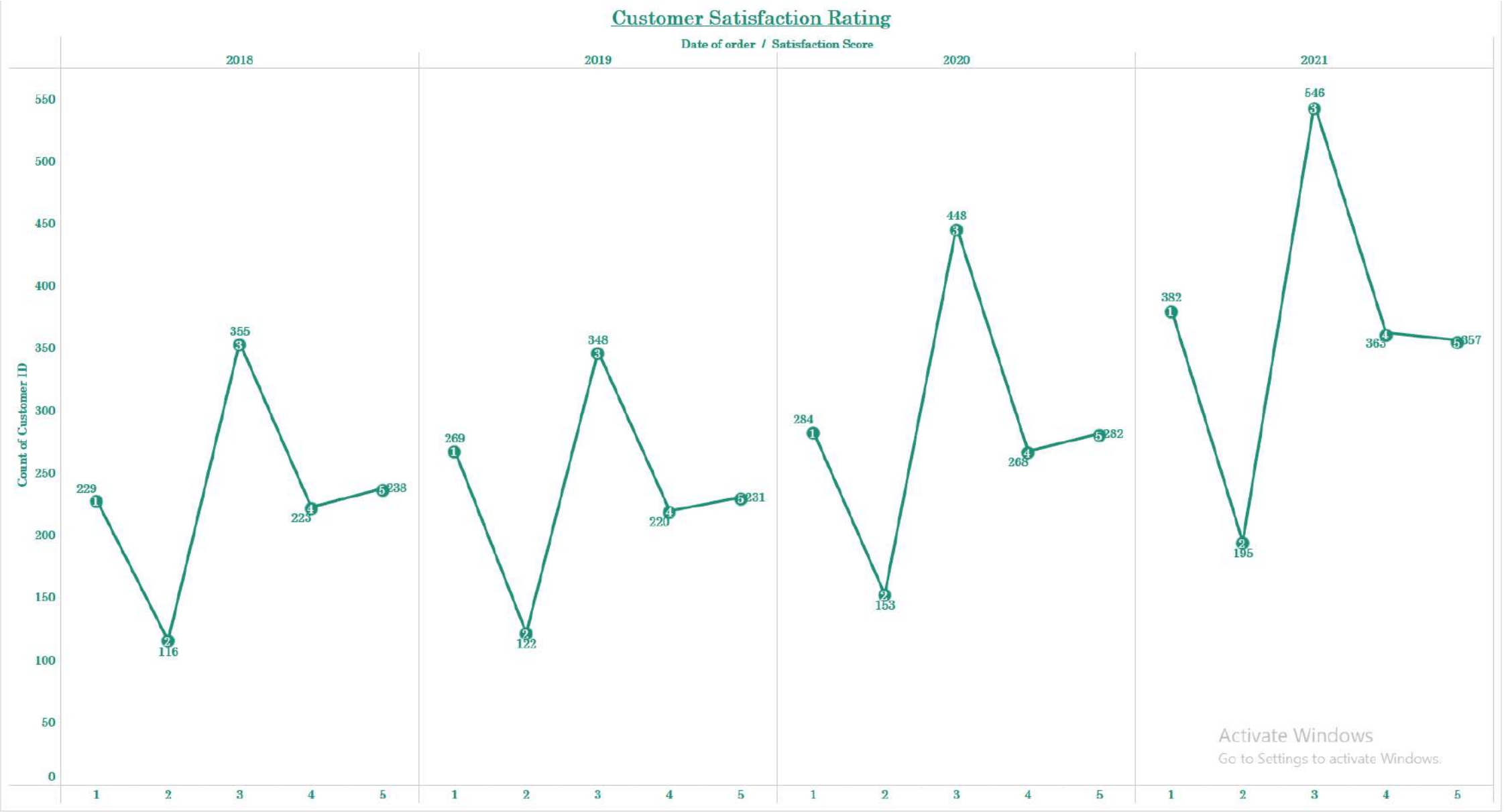


Activate Windows
Go to Settings to activate Windows.

Retention Rate Trend



Activate Windows
Go to Settings to activate Windows.

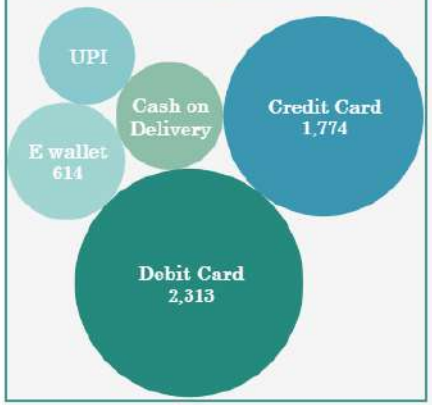

Customer Satisfaction Rating
Date of order / Satisfaction Score
2018
2019
2020
2021
Count of Customer ID
550
500
450
400
350
300
250
200
150
100
50
0
1
2
3
4
5
229
116
355
223
238
269
122
348
220
231
284
153
448
265
282
382
195
546
365
357
Activate Windows
Go to Settings to activate Windows.

RTP STORES

RTP STORES CUSTOMER RETENTION ANALYSIS

YEAR(Date of order)
Null

Preferred Payment Mode



Preferred Payment Mode (group)
Cash on De.. Credit Card Debit Card

₹ 997.61K
Total Cashback Amount

2.935
Average Time Spent on App

3.067
Average Satisfaction Score

9,921
Total Coupons Used

1,603
Complaints Attended

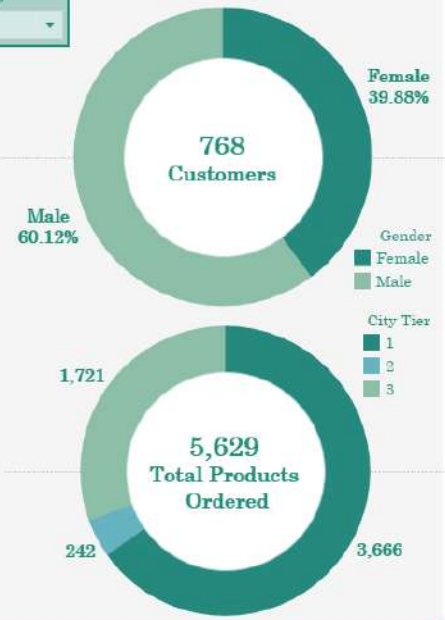
Computer

1,634

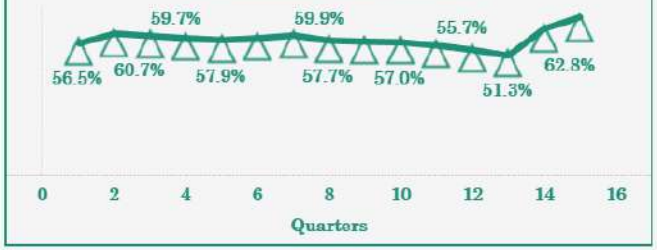
Mobile Phone

3,995

Profit Per Month



Retention Rate Trend



Champion Products

Products Delivered Vs Distance Graph

Activate Windows
Go to Settings to activate Windows.

Import necessary libraries and load the dataset

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('E_Commerce.csv')
```

In [2]:
```
df.head()
```

Out[2]:

	OrderID	CustomerID	Date of order	Date of Signup	Churn	PreferredLoginDevice	Purchased Product	Product Price	Cost Price	Profit	...	NumberOfDeviceRegi
0	IN-2020-152156	CG-12520	09-11-2020	08-11-2020	1	Mobile Phone	i-Phone 13	65999	56099.15	1319.98	...	
1	IN-2020-152156	CG-12520	09-11-2020	08-11-2020	1	Phone	i-Phone 14 Pro Max	129999	110499.15	2599.98	...	
2	IN-2020-138688	DV-13045	13-06-2020	12-06-2020	1	Phone	i-Phone 14 Pro Max	129999	110499.15	2599.98	...	
3	IN-2019-108966	SO-20335	11-10-2019	11-10-2019	1	Phone	i-Phone 14 Pro Max	129999	110499.15	2599.98	...	
4	IN-2019-108966	SO-20335	11-10-2019	11-10-2019	1	Phone	i-Phone 14 Pro Max	129999	110499.15	2599.98	...	

5 rows × 25 columns

Explore the data

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5630 entries, 0 to 5629
Data columns (total 25 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   OrderID                      5630 non-null   object 
 1   CustomerID                   5630 non-null   object 
 2   Date of order                5630 non-null   object 
 3   Date of Signup               5630 non-null   object 
 4   Churn                        5630 non-null   int64  
 5   PreferredLoginDevice         5630 non-null   object 
 6   Purchased Product            5630 non-null   object 
 7   Product Price                5630 non-null   int64  
 8   Cost Price                   5630 non-null   float64
 9   Profit                       5630 non-null   float64
 10  CityTier                     5630 non-null   int64  
 11  WarehouseToHome              5379 non-null   float64
 12  PreferredPaymentMode         5630 non-null   object 
 13  Gender                       5630 non-null   object 
 14  HourSpendOnApp               5375 non-null   float64
 15  NumberOfDeviceRegistered     5630 non-null   int64  
 16  PreferedOrderCat             5630 non-null   object 
 17  SatisfactionScore            5630 non-null   int64  
 18  MaritalStatus                5630 non-null   object 
 19  NumberOfAddress              5630 non-null   int64  
 20  Complain                     5630 non-null   int64  
 21  OrderAmountHikeFromlastYear  5365 non-null   float64
 22  CouponUsed                   5374 non-null   float64
 23  OrderCount                   5372 non-null   float64
 24  CashbackAmount               5630 non-null   int64  
dtypes: float64(7), int64(8), object(10)
memory usage: 1.1+ MB
```

In [4]:
```
df.describe()
```

Out[4]:

	Churn	Product Price	Cost Price	Profit	CityTier	WarehouseToHome	HourSpendOnApp	NumberOfDeviceR
count	5630.000000	5630.000000	5630.000000	5630.000000	5630.000000	5379.000000	5375.000000	56
mean	0.168384	59364.833037	50077.244076	5562.764018	1.654707	15.639896	2.931535	
std	0.374240	35660.648511	30404.510368	3771.860758	0.915389	8.531475	0.721926	
min	0.000000	9999.000000	8499.150000	999.900000	1.000000	5.000000	0.000000	
25%	0.000000	25999.000000	22099.150000	1799.900000	1.000000	9.000000	2.000000	
50%	0.000000	51999.000000	41599.000000	4599.900000	1.000000	14.000000	3.000000	
75%	0.000000	72999.000000	62049.150000	8999.900000	3.000000	20.000000	3.000000	
max	1.000000	169999.000000	144499.150000	12999.900000	3.000000	127.000000	5.000000	

Check for missing values

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
OrderID                        0
CustomerID                     0
Date of order                  0
Date of Signup                 0
Churn                          0
PreferredLoginDevice           0
Purchased Product              0
Product Price                  0
Cost Price                     0
Profit                         0
CityTier                       0
WarehouseToHome              251
PreferredPaymentMode           0
Gender                         0
HourSpendOnApp               255
NumberOfDeviceRegistered       0
PreferedOrderCat               0
SatisfactionScore              0
MaritalStatus                  0
NumberOfAddress                0
Complain                       0
OrderAmountHikeFromlastYear  265
CouponUsed                   256
OrderCount                   258
CashbackAmount                 0
dtype: int64
```

Filling the missing values with modal

In [6]:
```
for i in df.columns:
    df[i].fillna(df[i].mode()[0], inplace=True)
print(df)
```

```
            OrderID CustomerID Date of order Date of Signup  Churn  \
0     IN-2020-152156   CG-12520    09-11-2020     08-11-2020      1
1     IN-2020-152156   CG-12520    09-11-2020     08-11-2020      1
2     IN-2020-138688   DV-13045    13-06-2020     12-06-2020      1
3     IN-2019-108966   SO-20335    11-10-2019     11-10-2019      1
4     IN-2019-108966   SO-20335    11-10-2019     11-10-2019      1
...              ...        ...           ...            ...    ...
5625  IN-2018-126683   PP-18955    29-09-2018     29-09-2018      0
5626  IN-2018-126683   PP-18955    29-09-2018     29-09-2018      0
5627  IN-2021-148810   DR-12880    27-06-2021     26-06-2021      0
5628  IN-2021-148810   DR-12880    27-06-2021     26-06-2021      0
5629  IN-2020-146066   RB-19465    22-08-2020     21-08-2020      0

     PreferredLoginDevice   Purchased Product  Product Price  Cost Price  \
0            Mobile Phone          i-Phone 13          65999    56099.15
1                  Phone  i-Phone 14 Pro Max         129999   110499.15
2                  Phone  i-Phone 14 Pro Max         129999   110499.15
3                  Phone  i-Phone 14 Pro Max         129999   110499.15
4                  Phone  i-Phone 14 Pro Max         129999   110499.15
...                  ...                ...            ...         ...
5625            Computer         Vaio Laptop          79999    67999.15
5626         Mobile Phone          Asus Rog 3          34599    29409.15
5627         Mobile Phone          Asus Rog 3          34599    29409.15
5628            Computer         Vaio Laptop          79999    67999.15
5629         Mobile Phone          Asus Rog 3          34599    29409.15

       Profit  ...  NumberOfDeviceRegistered    PreferedOrderCat  \
0     1319.98  ...                         3  Laptop & Accessory
1     2599.98  ...                         4              Mobile
2     2599.98  ...                         4              Mobile
3     2599.98  ...                         4  Laptop & Accessory
4     2599.98  ...                         3              Mobile
...       ...  ...                       ...                 ...
5625  7999.90  ...                         2  Laptop & Accessory
5626  3459.90  ...                         5             Fashion
5627  3459.90  ...                         2  Laptop & Accessory
5628  7999.90  ...                         5  Laptop & Accessory
5629  3459.90  ...                         2  Laptop & Accessory

     SatisfactionScore MaritalStatus  NumberOfAddress  Complain  \
0                    2        Single                9         1
1                    3        Single                7         1
2                    3        Single                6         1
```

```
3                    5        Single               8         0
4                    5        Single               3         0
...                ...           ...             ...       ...
5625                 1       Married               6         0
5626                 5       Married               6         0
5627                 4       Married               3         1
5628                 4       Married               4         0
5629                 3       Married               4         0

     OrderAmountHikeFromlastYear  CouponUsed OrderCount  CashbackAmount
0                           11.0         1.0        1.0             160
1                           15.0         0.0        1.0             121
2                           14.0         0.0        1.0             120
3                           23.0         0.0        1.0             134
4                           11.0         1.0        1.0             130
...                          ...         ...        ...             ...
5625                        18.0         1.0        2.0             151
5626                        16.0         1.0        2.0             225
5627                        21.0         1.0        2.0             186
5628                        15.0         2.0        2.0             179
5629                        13.0         2.0        2.0             169

[5630 rows x 25 columns]
```

In [7]:
```
df.head()
```

Out[7]:

	OrderID	CustomerID	Date of order	Date of Signup	Churn	PreferredLoginDevice	Purchased Product	Product Price	Cost Price	Profit	...	NumberOfDeviceRegi
0	IN-2020-152156	CG-12520	09-11-2020	08-11-2020	1	Mobile Phone	i-Phone 13	65999	56099.15	1319.98	...	
1	IN-2020-152156	CG-12520	09-11-2020	08-11-2020	1	Phone	i-Phone 14 Pro Max	129999	110499.15	2599.98	...	
2	IN-2020-138688	DV-13045	13-06-2020	12-06-2020	1	Phone	i-Phone 14 Pro Max	129999	110499.15	2599.98	...	
3	IN-2019-108966	SO-20335	11-10-2019	11-10-2019	1	Phone	i-Phone 14 Pro Max	129999	110499.15	2599.98	...	
4	IN-2019-108966	SO-20335	11-10-2019	11-10-2019	1	Phone	i-Phone 14 Pro Max	129999	110499.15	2599.98	...	

5 rows × 25 columns

In [8]:
```
df.isnull().sum()
```

Out[8]:
```
OrderID                      0
CustomerID                   0
Date of order                0
Date of Signup               0
Churn                        0
PreferredLoginDevice         0
Purchased Product            0
Product Price                0
Cost Price                   0
Profit                       0
CityTier                     0
WarehouseToHome              0
PreferredPaymentMode         0
Gender                       0
HourSpendOnApp               0
NumberOfDeviceRegistered     0
PreferedOrderCat             0
SatisfactionScore            0
MaritalStatus                0
NumberOfAddress              0
Complain                     0
OrderAmountHikeFromlastYear  0
CouponUsed                   0
OrderCount                   0
CashbackAmount               0
dtype: int64
```

In [9]:
```
df["Churn"].value_counts()
```

Out[9]:
```
0    4682
1     948
Name: Churn, dtype: int64
```

In [10]:
```
df['PreferredLoginDevice'].unique()
```

Out[10]:
```
array(['Mobile Phone', 'Phone', 'Computer'], dtype=object)
```

In [11]:
```
df['PreferredLoginDevice']= df['PreferredLoginDevice'].str.replace('Mobile Phone','Phone')
df['PreferredLoginDevice']= df['PreferredLoginDevice'].str.replace('Phone','Mobile Phone')
df['PreferredLoginDevice']= df['PreferredLoginDevice'].str.replace('Computer','Computer')
```

In [12]:
```
df['PreferredLoginDevice'].value_counts()
```

Out[12]:
```
Mobile Phone    3996
Computer        1634
Name: PreferredLoginDevice, dtype: int64
```

In [13]:
```
df['PreferedOrderCat'].unique()
```

Out[13]:
```
array(['Laptop & Accessory', 'Mobile', 'Mobile Phone', 'Others',
       'Fashion', 'Grocery'], dtype=object)
```

In [22]:
```
df['PreferedOrderCat'].value_counts()
```

Out[22]:
```
Laptop & Accessory    2050
Mobile Phone Phone    1271
Fashion                826
Mobile Phone           809
Grocery                410
Others                 264
Name: PreferedOrderCat, dtype: int64
```

In [23]:
```
df['PreferedOrderCat']= df['PreferedOrderCat'].str.replace('Mobile Phone Phone','Mobile Phone')
df['PreferedOrderCat']= df['PreferedOrderCat'].str.replace('Mobile Phone','Mobile Phone')
```

In [24]:
```
df['PreferedOrderCat'].value_counts()
```

Out[24]:
```
Mobile Phone          2080
Laptop & Accessory    2050
Fashion                826
Grocery                410
Others                 264
Name: PreferedOrderCat, dtype: int64
```

In [55]:
```
df['PreferredPaymentMode'].unique()
```

Out[55]:
```
array(['Debit Card', 'UPI', 'Credit Card', 'Cash on Delivery', 'E wallet'],
      dtype=object)
```

In [26]:
```
df['PreferredPaymentMode']= df['PreferredPaymentMode'].str.replace('Debit Card','Debit Card')
df['PreferredPaymentMode']= df['PreferredPaymentMode'].str.replace('UPI','UPI')
df['PreferredPaymentMode']= df['PreferredPaymentMode'].str.replace('CC','Credit Card')
df['PreferredPaymentMode']= df['PreferredPaymentMode'].str.replace('Cash on Delivery','Cash on Delivery')
df['PreferredPaymentMode']= df['PreferredPaymentMode'].str.replace('E wallet','E wallet')
df['PreferredPaymentMode']= df['PreferredPaymentMode'].str.replace('COD','Cash on Delivery')
df['PreferredPaymentMode']= df['PreferredPaymentMode'].str.replace('Credit Card','Credit Card')
```

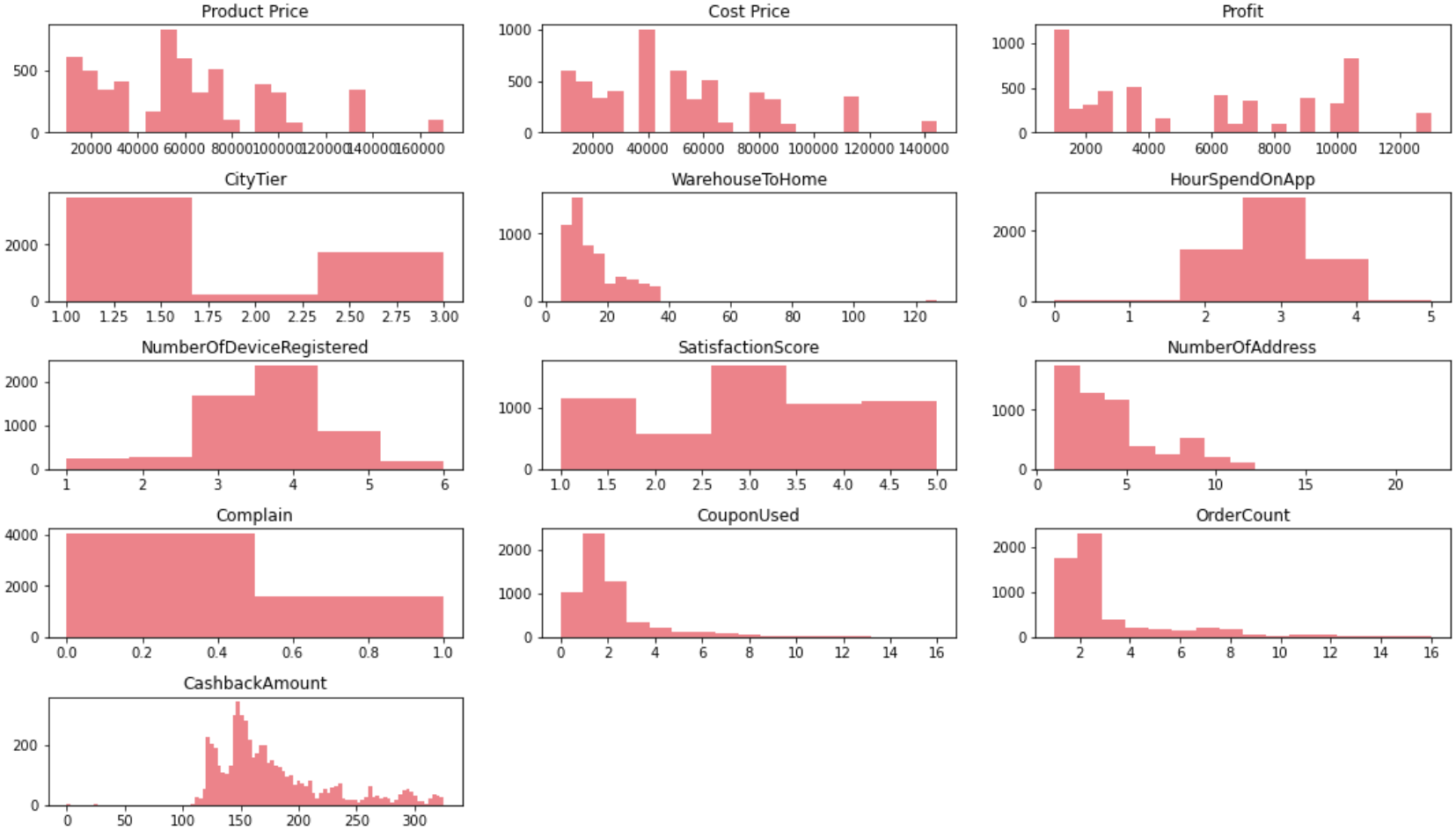
In [27]:
```
df['PreferredPaymentMode'].value_counts()
```

Out[27]:
```
Debit Card          2314
Credit Card         1774
E wallet             614
Cash on Delivery     514
UPI                  414
Name: PreferredPaymentMode, dtype: int64
```

Exploratory Data Analysis

In [29]:
```
dataset2 = df[['Product Price','Cost Price','Profit','CityTier','WarehouseToHome','HourSpendOnApp','NumberOfD
                    'SatisfactionScore','NumberOfAddress','Complain','CouponUsed','OrderCount',
                    'CashbackAmount']]

#Histogram:

fig = plt.figure(figsize=(15, 12))
plt.suptitle('Histograms of Numerical Columns\n',horizontalalignment="center",fontstyle = "normal", fontsize
for i in range(dataset2.shape[1]):
    plt.subplot(6, 3, i + 1)
    f = plt.gca()
    f.set_title(dataset2.columns.values[i])

    vals = np.size(dataset2.iloc[:, i].unique())
    if vals >= 100:
        vals = 100

    plt.hist(dataset2.iloc[:, i], bins=vals, color = '#ec838a')
plt.tight_layout(rect=[0, 0.03, 1, 0.95])
```

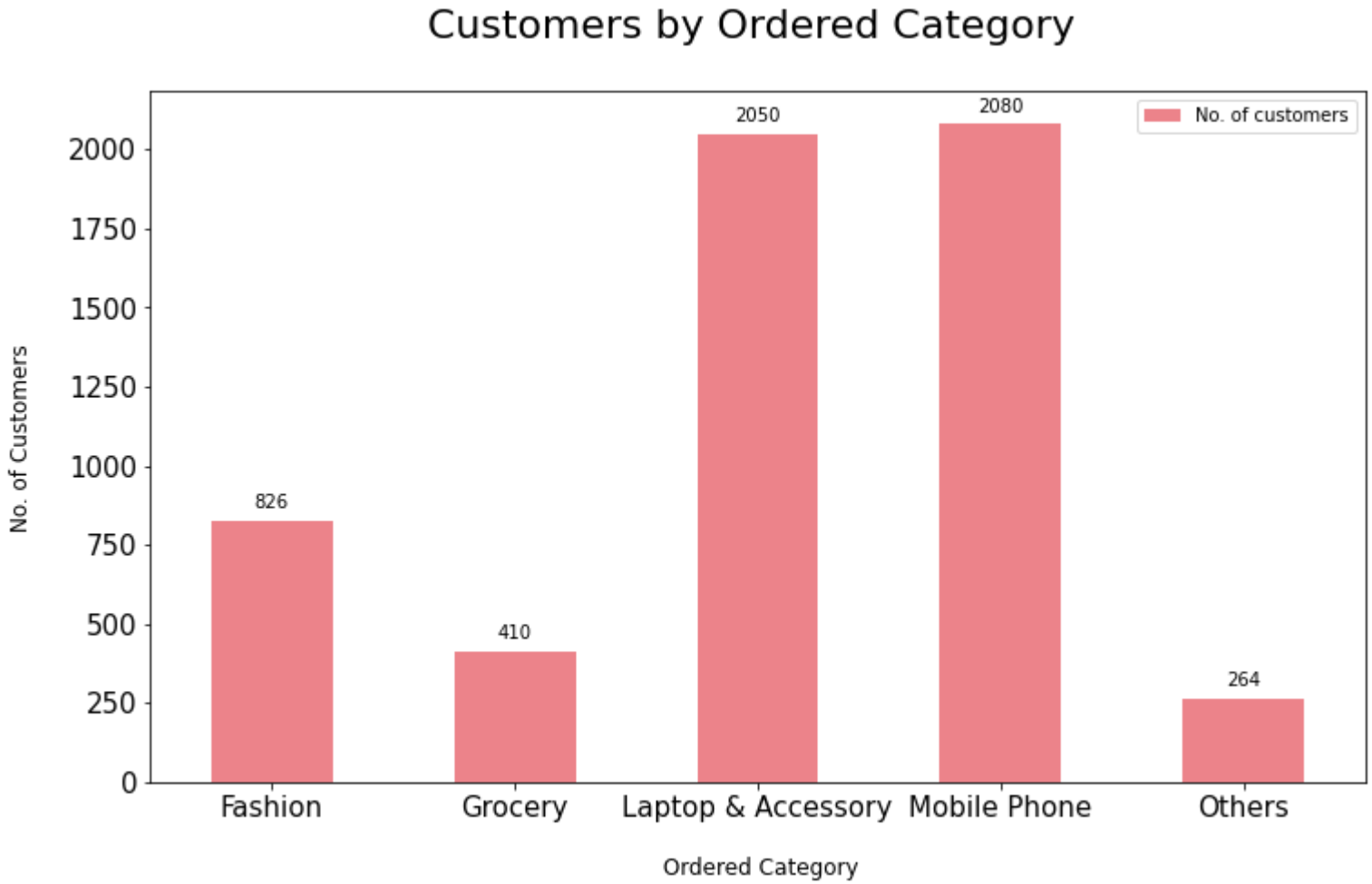
Histograms of Numerical Columns



Analyze distribution of Key Categorical Variables

In [30]:

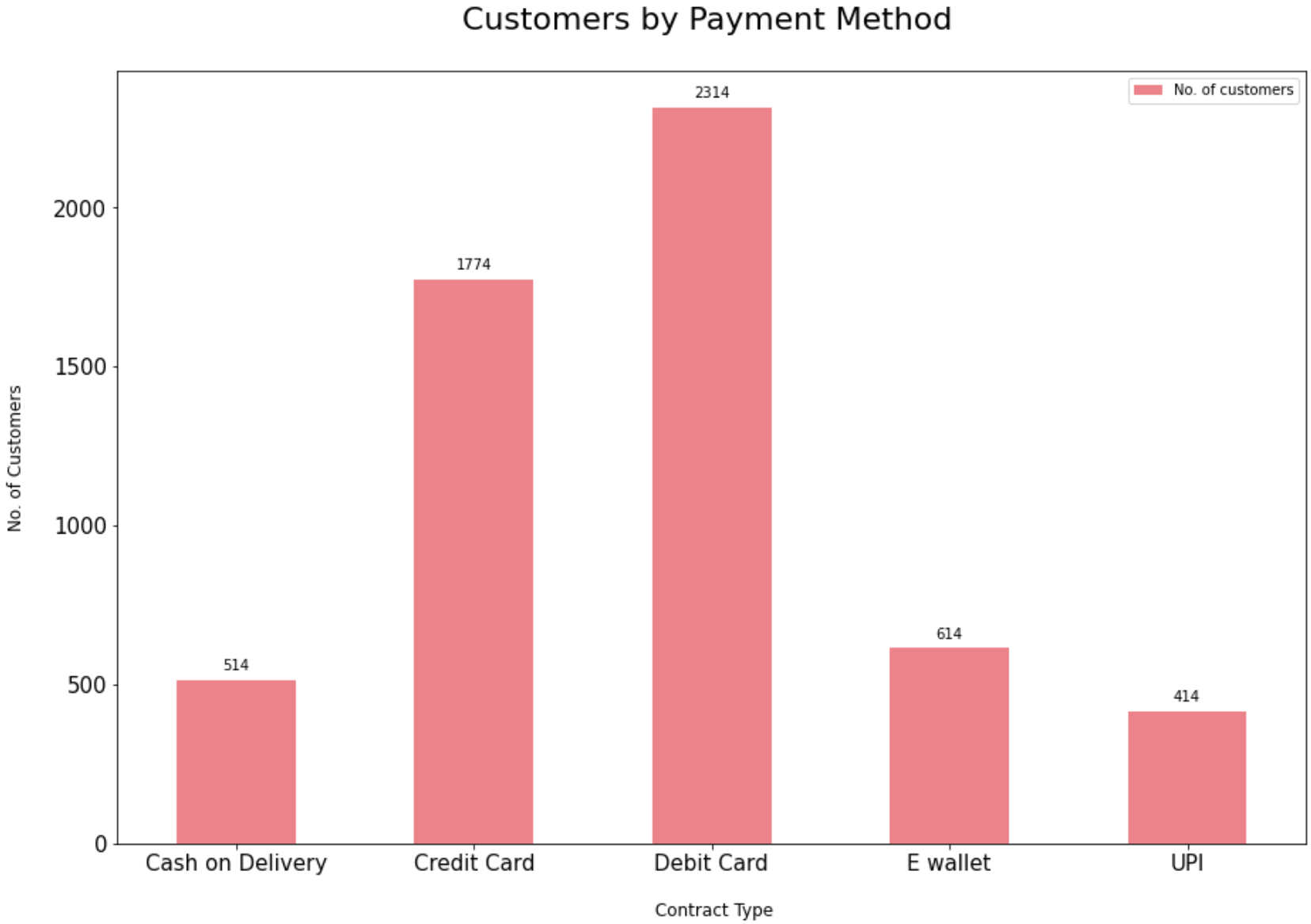
```
contract_split = df[[ "CustomerID", "PreferedOrderCat"]]
sectors = contract_split .groupby ("PreferedOrderCat")
contract_split = pd.DataFrame(sectors["CustomerID"].count())
contract_split.rename(columns={'CustomerID':'No. of customers'}, inplace=True)


ax =  contract_split[["No. of customers"]].plot.bar(title = 'Customers by Ordered Category', legend =True, ta

plt.ylabel('No. of Customers\n',horizontalalignment="center",fontstyle = "normal", fontsize = "large", fontfa
plt.xlabel('\n Ordered Category',horizontalalignment="center",fontstyle = "normal", fontsize = "large", fontf
plt.title('Customers by Ordered Category \n',horizontalalignment="center", fontstyle = "normal", fontsize = "
plt.legend(loc='upper right', fontsize = "medium")
plt.xticks(rotation=0, horizontalalignment="center")
plt.yticks(rotation=0, horizontalalignment="right")

x_labels = np.array(contract_split[["No. of customers"]])

def add_value_labels(ax, spacing=5):
    for rect in ax.patches:
        y_value = rect.get_height()
        x_value = rect.get_x() + rect.get_width() / 2
        space = spacing
        va = 'bottom'
        if y_value < 0:
            space *= -1
            va = 'top'
        label = "{:.0f}".format(y_value)
        ax.annotate(
            label,
            (x_value, y_value),
            xytext=(0, space),
            textcoords="offset points",
            ha='center',
            va=va)
add_value_labels(ax)
```


Customers by Ordered Category
No. of customers
No. of Customers
Ordered Category
0
250
500
750
1000
1250
1500
1750
2000
826
410
2050
2080
264
Fashion
Grocery
Laptop & Accessory
Mobile Phone
Others

Distribution of Payment Method Type

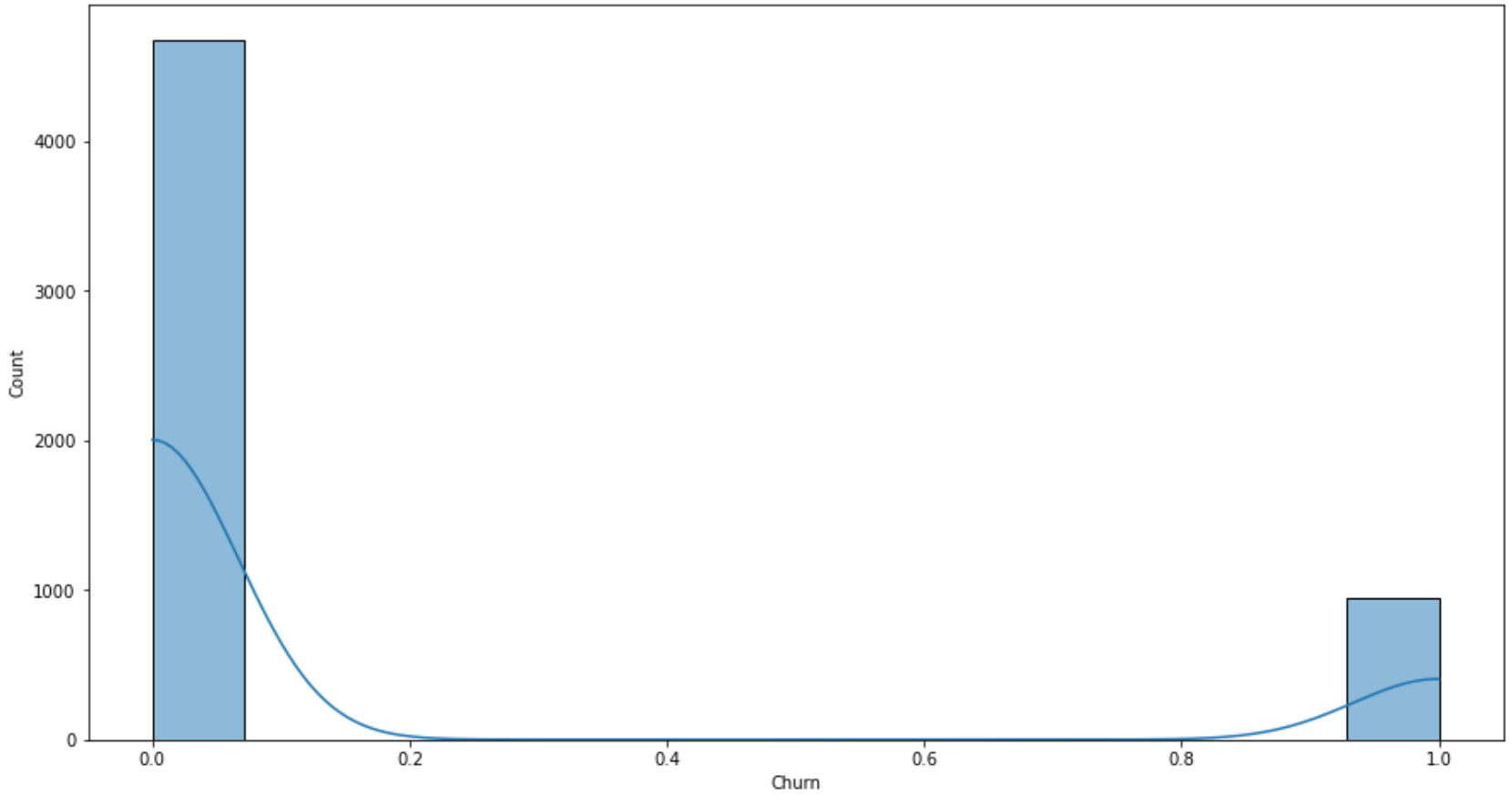
In [31]:

```
payment_method_split = df[[ "CustomerID", "PreferredPaymentMode"]]
sectors = payment_method_split  .groupby ("PreferredPaymentMode")
payment_method_split  = pd.DataFrame(sectors["CustomerID"].count())
payment_method_split.rename(columns={'CustomerID':'No. of customers'}, inplace=True)


ax =  payment_method_split [["No. of customers"]].plot.bar(title = 'Customers by Payment Method', legend =Tru

plt.ylabel('No. of Customers\n',horizontalalignment="center",fontstyle = "normal", fontsize = "large", fontfa
plt.xlabel('\n Contract Type',horizontalalignment="center",fontstyle = "normal", fontsize = "large", fontfami
plt.title('Customers by Payment Method \n',horizontalalignment="center", fontstyle = "normal", fontsize = "22
plt.legend(loc='upper right', fontsize = "medium")
plt.xticks(rotation=0, horizontalalignment="center")
plt.yticks(rotation=0, horizontalalignment="right")

x_labels = np.array(payment_method_split [["No. of customers"]])

def add_value_labels(ax, spacing=5):
    for rect in ax.patches:
        y_value = rect.get_height()
        x_value = rect.get_x() + rect.get_width() / 2
        space = spacing
        va = 'bottom'
        if y_value < 0:
            space *= -1
            va = 'top'
        label = "{:.0f}".format(y_value)
        ax.annotate(
            label,
            (x_value, y_value),
            xytext=(0, space),
            textcoords="offset points",
            ha='center',
            va=va)
add_value_labels(ax)
```


Customers by Payment Method
No. of customers
2314
1774
514
614
414
2000
1500
1000
500
0
No. of Customers
Cash on Delivery
Credit Card
Debit Card
E wallet
UPI
Contract Type

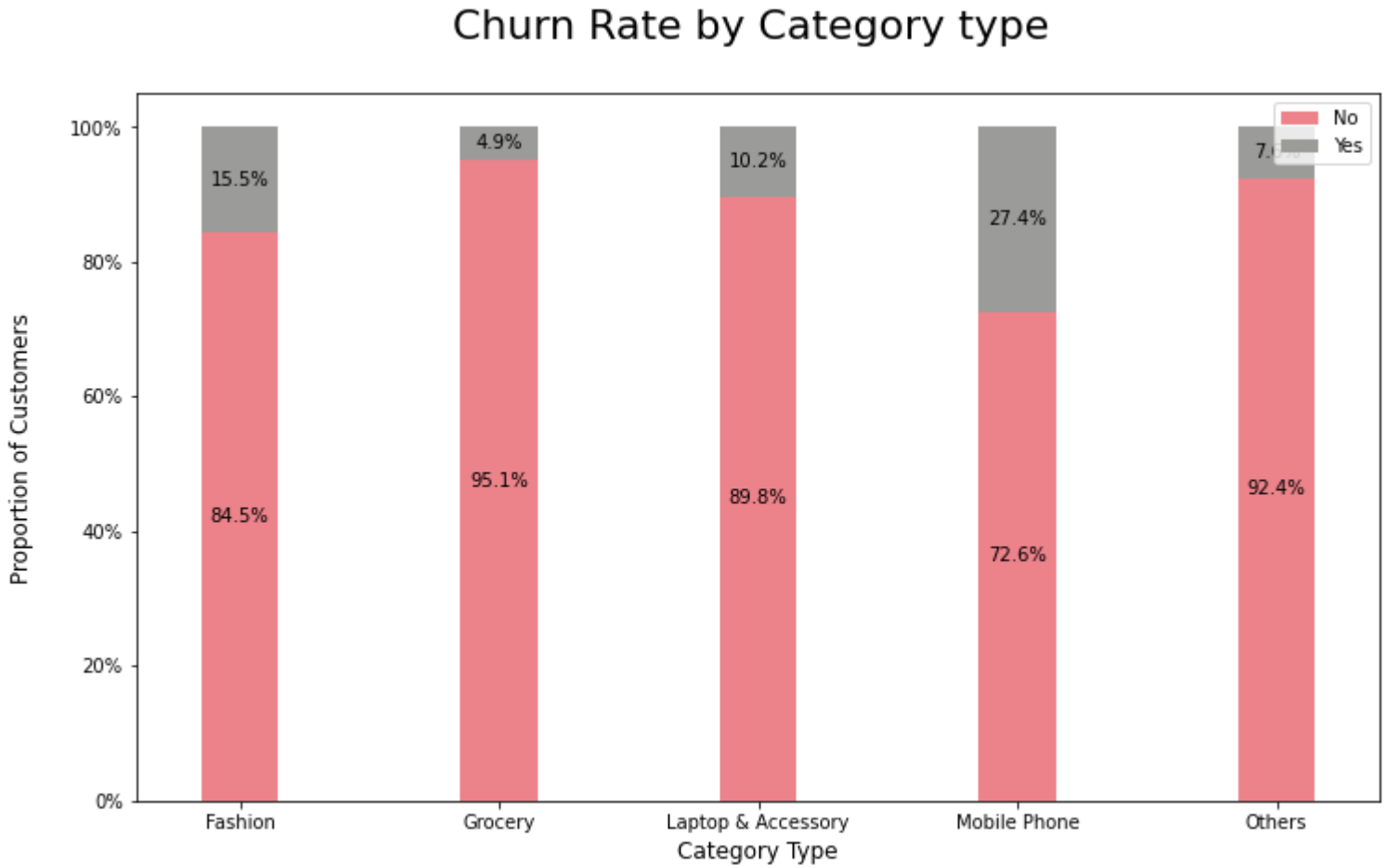
Analyze the overall churn rate

In [32]:
```
plt.figure(figsize=(15,8))
sns.histplot(x='Churn', data=df,kde=True)
plt.show()
```



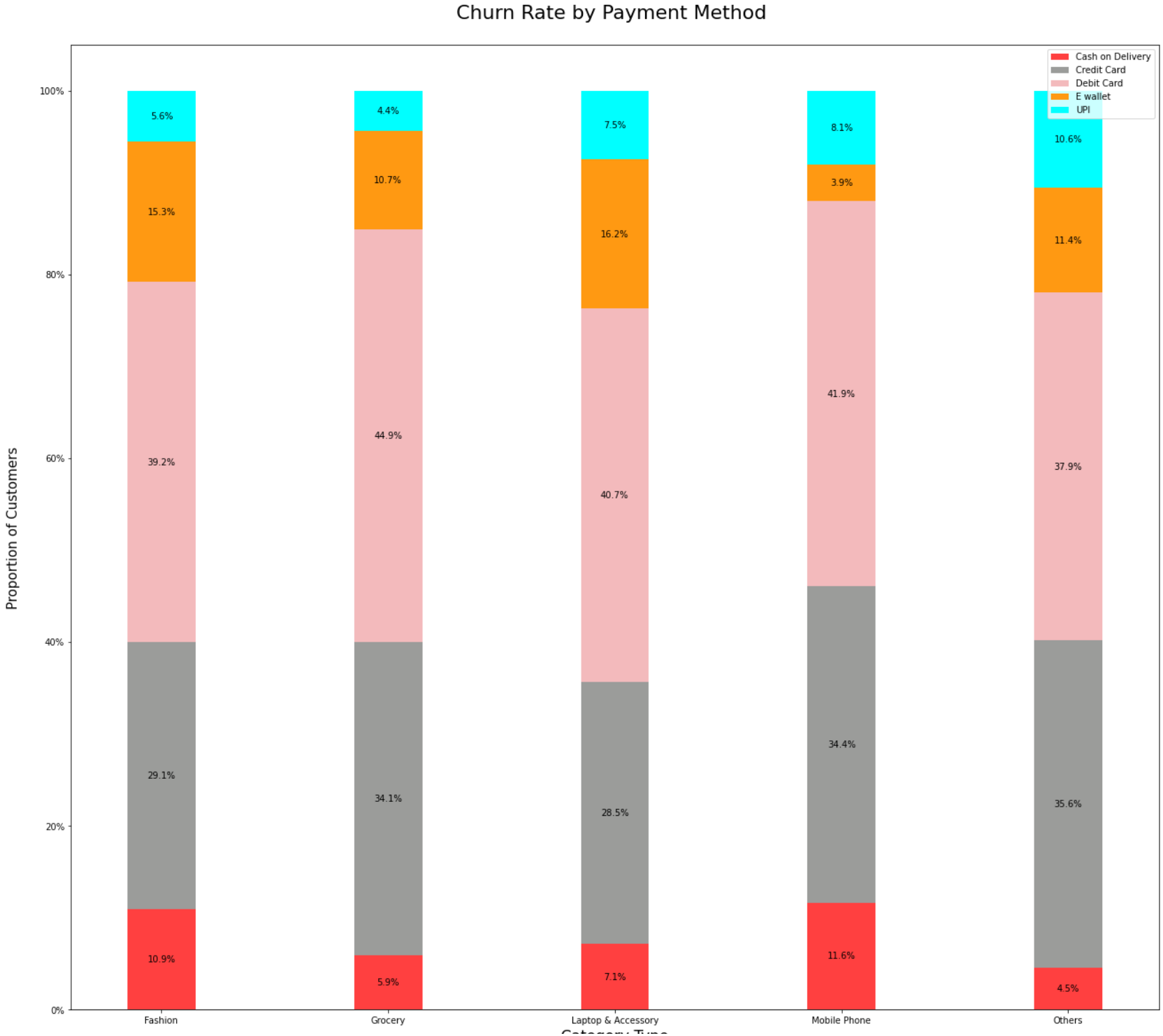
Churn Rate by Category Type

In [33]:
```
import matplotlib.ticker as mtick

contract_churn = df.groupby(['PreferedOrderCat','Churn']).size().unstack()

contract_churn.rename(columns={0:'No', 1:'Yes'}, inplace=True)

colors  = ['#ec838a','#9b9c9a']

ax = (contract_churn.T*100.0 / contract_churn.T.sum()).T.plot(kind='bar',
                                                              width = 0.3,
                                                              stacked = True,
                                                              rot = 0,
                                                              figsize = (12,7),
                                                              color = colors)




plt.ylabel('Proportion of Customers\n',horizontalalignment="center",fontstyle = "normal", fontsize = "large",
plt.xlabel('Category Type\n',horizontalalignment="center",fontstyle = "normal", fontsize = "large", fontfamil
plt.title('Churn Rate by Category type \n',horizontalalignment="center", fontstyle = "normal", fontsize = "22
plt.legend(loc='upper right', fontsize = "medium")
plt.xticks(rotation=0, horizontalalignment="center")
plt.yticks(rotation=0, horizontalalignment="right")
ax.yaxis.set_major_formatter(mtick.PercentFormatter())
for p in ax.patches:
    width, height = p.get_width(), p.get_height()
    x, y = p.get_xy()
    ax.text(x+width/2,
            y+height/2,
            '{:.1f}%'.format(height),
            horizontalalignment='center',
            verticalalignment='center')
ax.autoscale(enable=False, axis='both', tight=False)
```


Churn Rate by Category type
No
Yes
100%
80%
60%
40%
20%
0%
Proportion of Customers
15.5%
84.5%
4.9%
95.1%
10.2%
89.8%
27.4%
72.6%
7.6%
92.4%
Fashion
Grocery
Laptop & Accessory
Mobile Phone
Others
Category Type

Churn Rate by Payment Method

In [34]:
```
import matplotlib.ticker as mtick

contract_churn = df.groupby(['PreferedOrderCat','PreferredPaymentMode']).size().unstack()

contract_churn.rename(columns={0:'No', 1:'Yes'}, inplace=True)

colors  = ['#FF4040','#9b9c9a', '#f3babc' , '#FF9912', '#00FFFF','#E3CF57','#8A2BE2']

ax = (contract_churn.T*100.0 / contract_churn.T.sum()).T.plot(kind='bar',
                                                              width = 0.3,
                                                              stacked = True,
                                                              rot = 0,
                                                              figsize = (22,20),
                                                              color = colors)




plt.ylabel('Proportion of Customers\n',horizontalalignment="center",fontstyle = "normal", fontsize = "15", fo
plt.xlabel('Category Type\n',horizontalalignment="center",fontstyle = "normal", fontsize = "17", fontfamily =
plt.title('Churn Rate by Payment Method \n',horizontalalignment="center", fontstyle = "normal", fontsize = "2
plt.legend(loc='best', fontsize = "medium")
plt.xticks(rotation=0, horizontalalignment="center")
plt.yticks(rotation=0, horizontalalignment="right")
ax.yaxis.set_major_formatter(mtick.PercentFormatter())
for p in ax.patches:
    width, height = p.get_width(), p.get_height()
    x, y = p.get_xy()
    ax.text(x+width/2,
            y+height/2,
            '{:.1f}%'.format(height),
            horizontalalignment='center',
            verticalalignment='center')
ax.autoscale(enable=False, axis='both', tight=False)
```


Churn Rate by Payment Method
Cash on Delivery
Credit Card
Debit Card
E wallet
UPI
Proportion of Customers
100%
80%
60%
40%
20%
0%
5.6%
15.3%
39.2%
29.1%
10.9%
4.4%
10.7%
44.9%
34.1%
5.9%
7.5%
16.2%
40.7%
28.5%
7.1%
8.1%
3.9%
41.9%
34.4%
11.6%
10.6%
11.4%
37.9%
35.6%
4.5%
Fashion
Grocery
Laptop & Accessory
Mobile Phone
Others

Category Type

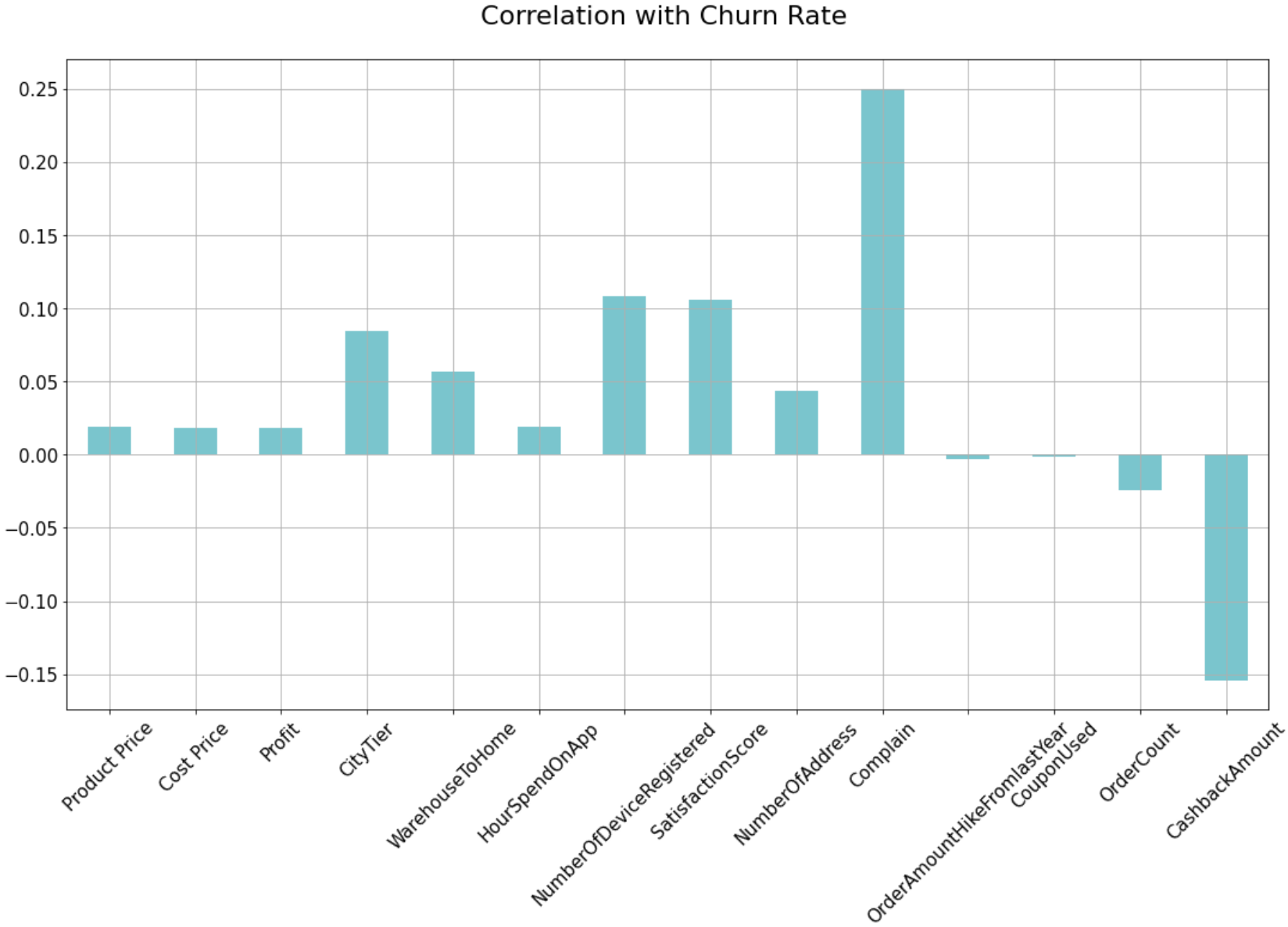
Finding positive and negative correlations

In [35]:
```
dataset2 = df[['Product Price', 'Cost Price', 'Profit',
       'CityTier', 'WarehouseToHome', 'HourSpendOnApp',
        'NumberOfDeviceRegistered', 'SatisfactionScore','NumberOfAddress','Complain','OrderAmountHikeFromlast
        'CouponUsed','OrderCount','CashbackAmount']]
correlations = dataset2.corrwith(df.Churn)
correlations = correlations[correlations!=1]
positive_correlations = correlations[
correlations >0].sort_values(ascending = False)
negative_correlations =correlations[
correlations<0].sort_values(ascending = False)
print('Most Positive Correlations: \n', positive_correlations)
print('\nMost Negative Correlations: \n', negative_correlations)
```

```
Most Positive Correlations:
 Complain                    0.250188
NumberOfDeviceRegistered    0.107939
SatisfactionScore           0.105481
CityTier                    0.084703
WarehouseToHome             0.056958
NumberOfAddress             0.043931
Product Price               0.019009
HourSpendOnApp              0.018816
Profit                      0.018153
Cost Price                  0.017919
dtype: float64

Most Negative Correlations:
 CouponUsed                   -0.001430
OrderAmountHikeFromlastYear   -0.003014
OrderCount                    -0.024038
CashbackAmount                -0.154161
dtype: float64
```

In [36]:
```
correlations = dataset2.corrwith(df.Churn)
correlations = correlations[correlations!=1]
correlations.plot.bar(
        figsize = (18, 10),
        fontsize = 15,
        color = '#7AC5CD',
        rot = 45, grid = True)
plt.title('Correlation with Churn Rate \n',
horizontalalignment="center", fontstyle = "normal",
fontsize = "22", fontfamily = "sans-serif")
```

Out[36]:
```
Text(0.5, 1.0, 'Correlation with Churn Rate \n')
```

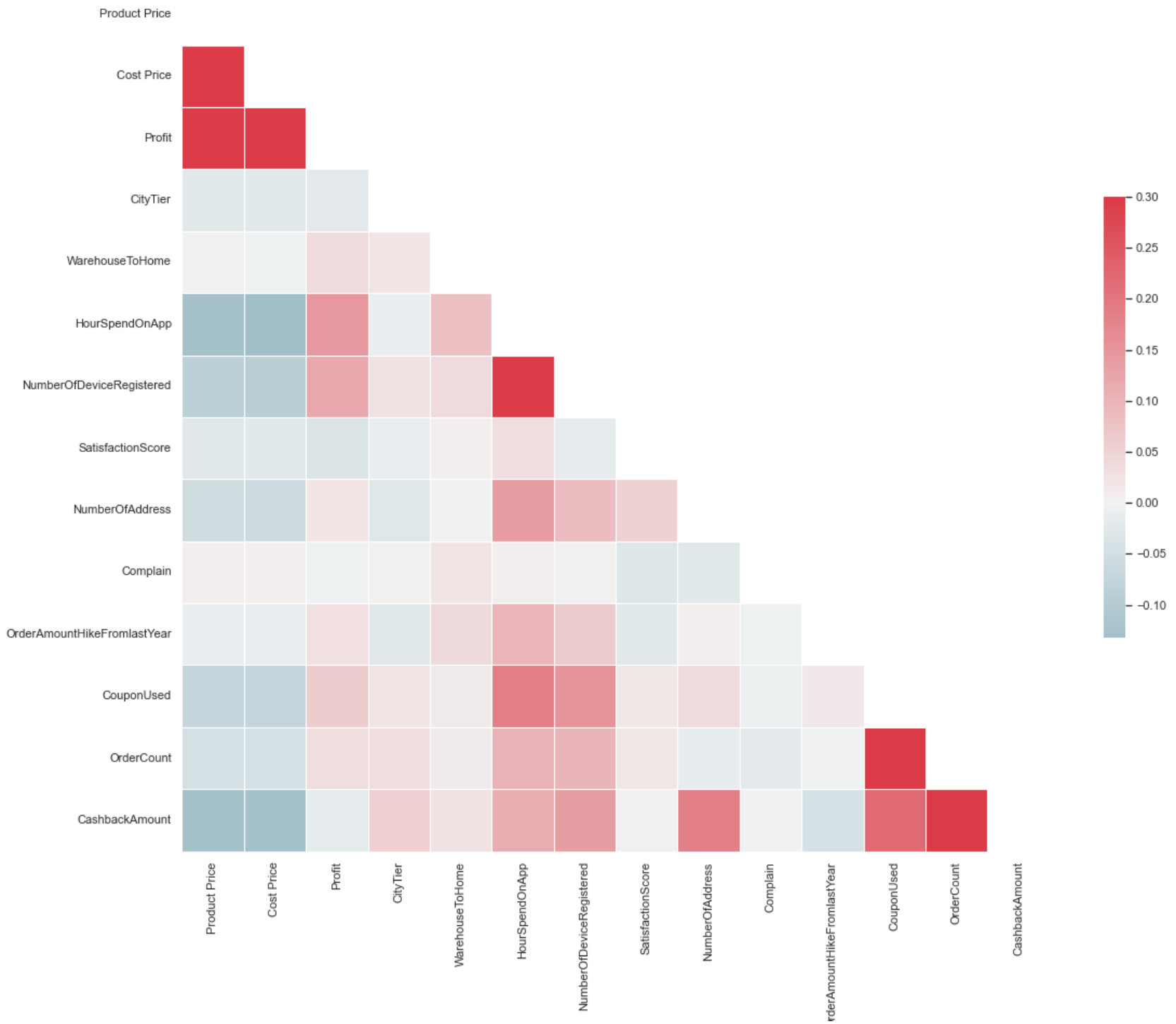

Correlation with Churn Rate
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
Product Price
Cost Price
Profit
CityTier
WarehouseToHome
HourSpendOnApp
NumberOfDeviceRegistered
SatisfactionScore
NumberOfAddress
Complain
OrderAmountHikeFromlastYear
CouponUsed
OrderCount
CashbackAmount

Plotting Correlation Matrix of all independent variables

In [37]:
```
## Set and compute the Correlation Matrix
sns.set(style="white")
corr = dataset2.corr()

# Generate a mask for the upper triangle
mask = np.zeros_like(corr, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True

# Set up the matplotlib figure and a diverging colormap
f, ax = plt.subplots(figsize=(18, 15))
cmap = sns.diverging_palette(220, 10, as_cmap=True)

# Draw the heatmap with the mask and correct aspect ratio
sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.3, center=0,
            square=True, linewidths=.5, cbar_kws={"shrink": .5})
```

```
C:\Users\HP\AppData\Local\Temp/ipykernel_3536/2346922627.py:6: DeprecationWarning: `np.bool` is a deprecated
alias for the builtin `bool`. To silence this warning, use `bool` by itself. Doing this will not modify any
behavior and is safe. If you specifically wanted the numpy scalar type, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html
#deprecations (https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations)
  mask = np.zeros_like(corr, dtype=np.bool)
```

Out[37]:
```
<AxesSubplot:>
```


Product Price
Cost Price
Profit
CityTier
WarehouseToHome
HourSpendOnApp
NumberOfDeviceRegistered
SatisfactionScore
NumberOfAddress
Complain
OrderAmountHikeFromlastYear
CouponUsed
OrderCount
CashbackAmount
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10

In [38]:
```
### Saving the file
```

In [39]:
```
df.to_excel('updated rrd.xlsx')
```

Label Encoding

In [40]:
```
identity = df["CustomerID"]

dataset = df.drop("CustomerID", axis=1)

# convert rest of categorical variable into dummy

dataset= pd.get_dummies(dataset)

#Rejoin userid to dataset (column concatenation)

dataset = pd.concat([dataset, identity], axis = 1)
```

Splitting the dataset into dependent and independent variables

In [41]:
```
response = dataset["Churn"]
dataset = dataset.drop(columns="Churn")
```

Generating training and test datasets

In [42]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(dataset, response,
                                                    stratify=response,
                                                    test_size = 0.2,random_state = 0)
```

In [43]:
```
print("Number transactions X_train dataset: ", X_train.shape)
print("Number transactions y_train dataset: ", y_train.shape)
print("Number transactions X_test dataset: ", X_test.shape)
print("Number transactions y_test dataset: ", y_test.shape)
```

```
Number transactions X_train dataset:  (4504, 4981)
Number transactions y_train dataset:  (4504,)
Number transactions X_test dataset:  (1126, 4981)
Number transactions y_test dataset:  (1126,)
```

Removing Identifiers

In [44]:
```
train_identity = X_train["CustomerID"]
X_train = X_train.drop(columns = ["CustomerID"])

test_identity = X_test["CustomerID"]
X_test = X_test.drop(columns = ["CustomerID"])
```

Feature Scaling

In [45]:
```
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train2 = pd.DataFrame(sc_X.fit_transform(X_train))
X_train2.columns = X_train.columns.values
X_train2.index = X_train.index.values
X_train = X_train2

X_test2 = pd.DataFrame(sc_X.transform(X_test))
X_test2.columns = X_test.columns.values
X_test2.index = X_test.index.values
X_test = X_test2
X_train[150:160]
```

Out[45]:

	Product Price	Cost Price	Profit	CityTier	WarehouseToHome	HourSpendOnApp	NumberOfDeviceRegistered	SatisfactionScore	N
1915	1.374987	1.383055	-0.891444	-0.718588	1.150766	-1.323938	-0.662196	-1.494556	
4881	0.842952	0.852670	0.911037	-0.718588	1.866756	1.525091	0.307733	-0.041457	
5406	0.564399	0.574982	0.646663	0.373355	-0.042550	1.525091	1.277661	-1.494556	
752	0.063004	0.075143	0.170789	-0.718588	-0.877872	-1.323938	-0.662196	-0.768007	
5338	0.842952	0.852670	0.911037	1.465299	-0.758540	0.100576	0.307733	-0.041457	
1824	1.374987	1.383055	-0.891444	-0.718588	-0.281214	0.100576	-0.662196	-0.041457	
133	1.957162	1.963423	-0.780936	-0.718588	-1.116535	0.100576	-0.662196	-0.768007	
4665	-1.385470	-1.368836	-1.203956	-0.718588	-0.281214	1.525091	0.307733	-1.494556	
2350	-1.051207	-1.035610	-0.886707	0.373355	0.912103	-1.323938	0.307733	1.411642	
4944	0.842952	0.852670	0.911037	-0.718588	1.031434	1.525091	1.277661	-0.041457	

10 rows × 4980 columns

Implementing Machine Learning Models

SVM Model

In [46]:
```
from sklearn import svm

svm.SVC(kernel='linear',gamma='auto',C=2,probability=True)

clfi=svm.SVC(kernel='linear',gamma='auto',C=2,probability=True)

clfi.fit(X_train,y_train)

clfi.score(X_train,y_train)

y_pred=clfi.predict(X_test)

y_pred
```

Out[46]:
```
array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

In [47]:
```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test, y_pred))
```

```
0.8179396092362344
```

In [48]:
```
from sklearn.metrics import confusion_matrix
print(confusion_matrix(y_test, y_pred))
```

```
[[829 107]
 [ 98  92]]
```

In [49]:
```
from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.89      0.89      0.89       936
           1       0.46      0.48      0.47       190

    accuracy                           0.82      1126
   macro avg       0.68      0.68      0.68      1126
weighted avg       0.82      0.82      0.82      1126
```

Naive Bayes

In [51]:
```
from sklearn.naive_bayes import GaussianNB
clfi=GaussianNB()
clfi.fit(X_train, y_train)
clfi.score(X_train, y_train)
y_pred = clfi.predict(X_test)
y_pred
```

Out[51]:
```
array([0, 0, 1, ..., 1, 0, 1], dtype=int64)
```

In [52]:
```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test, y_pred))
```

```
0.5719360568383659
```

In [53]:
```
from sklearn.metrics import confusion_matrix
print(confusion_matrix(y_test, y_pred))
```

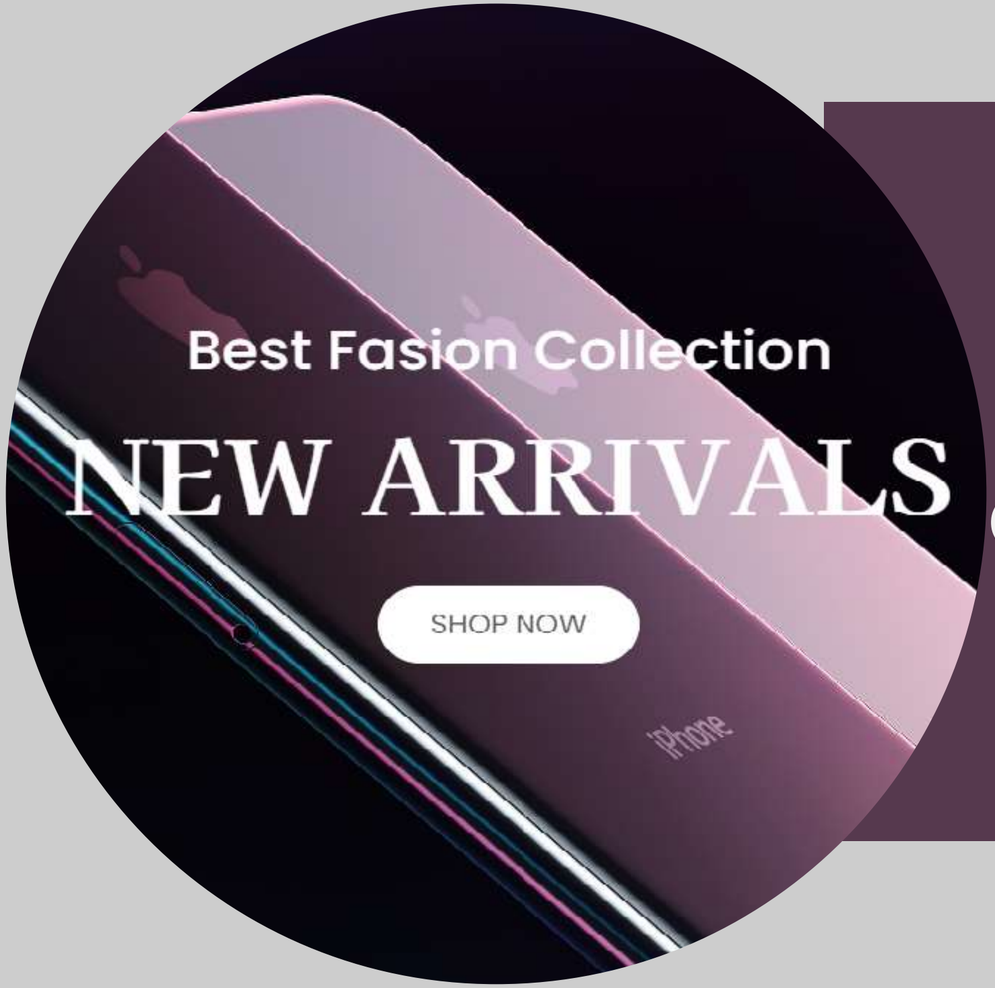
```
[[565 371]
 [111  79]]
```

In [54]:
```
from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.84      0.60      0.70       936
           1       0.18      0.42      0.25       190

    accuracy                           0.57      1126
   macro avg       0.51      0.51      0.47      1126
weighted avg       0.72      0.57      0.62      1126
```

In [ ]:

CLASS PROJECT



RTP STORES

CUSTOMER RETENTION ANALYSIS

Presented By:

DA/DS BATCH 4

ABSTRACT

Customer retention is a crucial aspect of business success, with direct impacts on a company's profitability and long-term sustainability. This group project is aimed to investigate customer retention strategies and practices in the context of **E-commerce website (RTP STORES)** . The project explored various approaches and techniques used by companies to retain their customers, including **loyalty programs, personalized marketing, and exceptional customer service**.

The project utilized a combination of **qualitative and quantitative research** methods, including **literature review and data analysis**, to examine the effectiveness of different customer retention strategies. The findings revealed key factors influencing customer retention, such as **customer satisfaction, trust, and loyalty**. The project also identified challenges and opportunities in implementing customer retention strategies, including resource allocation, data management, and customer segmentation.

To summarize, this group project contributes to the understanding of customer retention in the context of E-commerce website (RTP STORES) and provides practical recommendations for businesses to enhance their customer retention efforts and achieve sustainable business growth.

INTRODUCTION TO RTP STORES

RTP Stores is a leading e-commerce company that specializes in providing high-quality products to customers worldwide. Our mission is to offer a seamless shopping experience through our user-friendly website and mobile app, with a wide range of products to meet the diverse needs of our customers.

At RTP Stores, we pride ourselves on our commitment to customer satisfaction. We understand that today's consumers have high expectations when it comes to online shopping, which is why we go above and beyond to provide a personalized and efficient service. Our team is dedicated to ensuring that every customer enjoys a hassle-free shopping experience, from browsing our products to placing an order and receiving their delivery.

With a vast selection of products, from electronics and gadgets to fashion and beauty items, RTP Stores is your one-stop shop for all your online shopping needs. Our product range is constantly evolving to keep up with the latest trends and consumer demands, ensuring that our customers always have access to the newest and most innovative products on the market.

In addition to our commitment to customer satisfaction, RTP Stores is also committed to sustainability and social responsibility. We strive to minimize our environmental impact by adopting sustainable practices throughout our operations, and we also support charitable initiatives that help to make a positive difference in the world.
We are proud to be a part of the e-commerce industry, and we are dedicated to delivering the best possible online shopping experience to our customers. Thank you for choosing RTP Stores for your online shopping needs.

WEBSITERTP Stores_index.html

PROBLEM STATEMENT

- **RTP stores** has noticed a decreasing trend in customer loyalty over the past year, which is affecting the company's revenue and market share.

- The objective of this project is to analyze customer behavior and preferences, identify factors that contribute to customer disloyalty, and develop targeted retention strategies to improve customer satisfaction and loyalty.

OBJECTIVES

- To identify the factors contributing to the churn rate of the customers of electronics section of the company.
- To identify key factors influencing customer retention, such as customer satisfaction.
- To explore various approaches and techniques used by companies to retain their customers, including cashback loyalty programs, personalized marketing, and exceptional customer service.
- To provide practical recommendations for businesses seeking to enhance their customer retention efforts and achieve sustainable business growth

INTRODUCTION TO CUSTOMER RETENTION

Customer retention is a critical aspect of business success, and in today's competitive landscape, companies are constantly seeking ways to retain their existing customers and build long-term relationships with them. This group project on customer retention aims to investigate effective strategies and practices in the context of E-commerce website (RTP STORES). The project involved a multidisciplinary team of experts, including those responsible for data collection, data processing, visualization, and front-end development.

Acquiring a new customer can cost five times more than retaining an existing customer. Increasing customer retention by 5% can increase profits from 25-95%. The success rate of selling to a customer you already have is 60-70%, while the success rate of selling to a new customer is 5-20%. They also have a higher average order value than first-time consumers. By contributing to the understanding of customer retention, this project aims to assist businesses in developing effective strategies to retain customers and build lasting relationships with them.

WHAT IS CUSTOMER RETENTION AND CUSTOMER CHURN ?

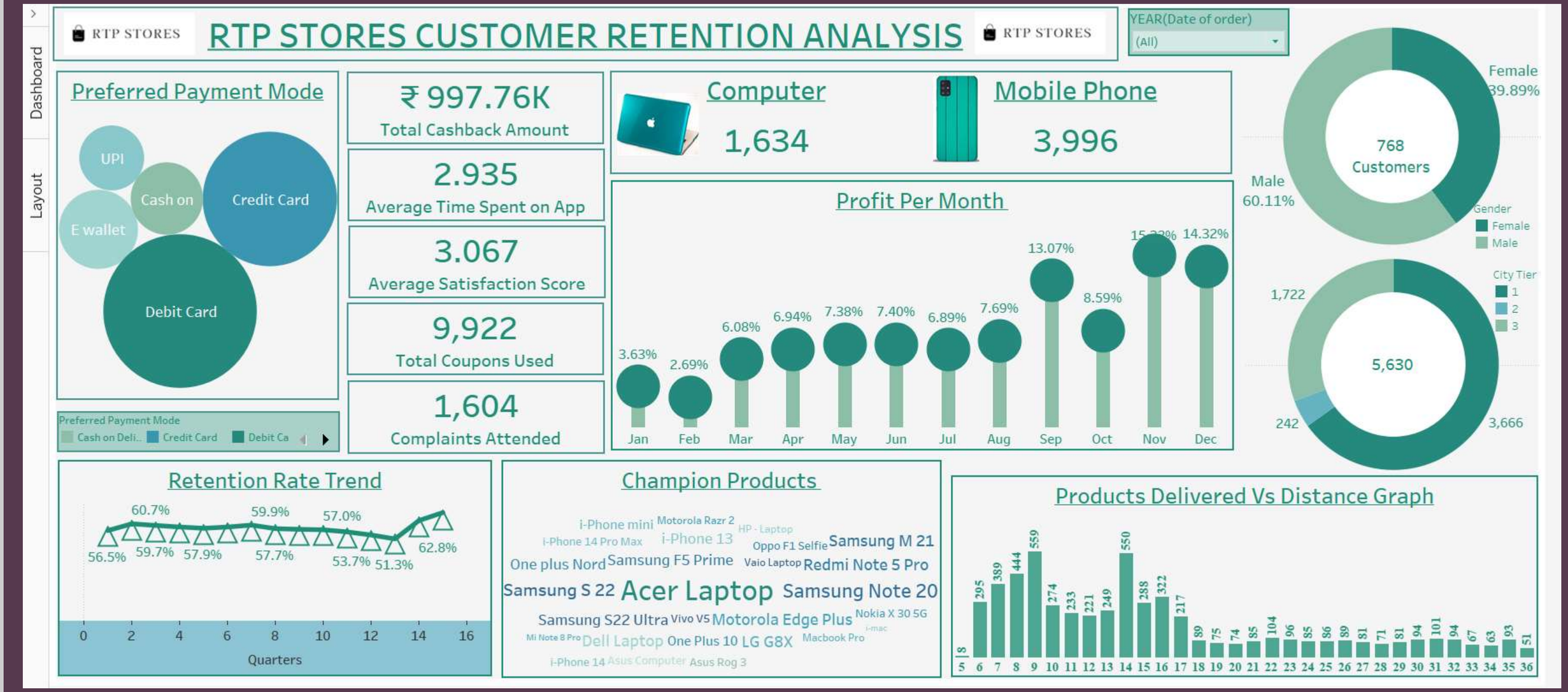
- **Customer retention** refers to the ability of a business to retain its existing customers over a period of time. It involves building long-term relationships with customers by providing high-quality products or services, excellent customer service, and personalized experiences that meet their needs and expectations. Customer retention is important for businesses because it can lead to increased customer loyalty, repeat purchases, positive word-of-mouth referrals, and ultimately, sustained revenue growth.
- On the other hand, **Customer churn** refers to the rate at which customers discontinue using a product or service. It reflects the number of customers who cancel their subscriptions, switch to a competitor, or simply stop using a product or service over a given time period. High churn rates can be a cause for concern for businesses because they indicate a lack of customer loyalty or satisfaction, and can lead to a decline in revenue and market share. Therefore, reducing churn and retaining more customers is a key objective for businesses that want to achieve sustainable growth.

EXPLORATORY DATA ANALYSIS (EDA)

- Exploratory Data Analysis (EDA) is a critical first step in any data analysis project. EDA involves examining and visualizing the data to identify patterns, trends, and relationships between variables. The team conducted EDA on the customer retention data set using Python programming language.
- The EDA included the following steps:
  - Data cleaning: The team removed any missing or duplicate data points and corrected any inconsistencies in the data.
  - Descriptive statistics: The team calculated summary statistics, such as mean, median, and standard deviation, for key variables to gain a better understanding of the data distribution.
  - Data visualization: The team created a range of visualizations to get the insights from the data set.

(PYTHON PRESENTATION)

DASHBOARD


Dashboard
Layout
RTP STORES
RTP STORES CUSTOMER RETENTION ANALYSIS
RTP STORES
YEAR(Date of order)
(All)
Preferred Payment Mode
UPI
Cash on
Credit Card
E wallet
Debit Card
Preferred Payment Mode
Cash on Deli..
Credit Card
Debit Ca
₹ 997.76K
Total Cashback Amount
2.935
Average Time Spent on App
3.067
Average Satisfaction Score
9,922
Total Coupons Used
1,604
Complaints Attended
Computer
1,634
Mobile Phone
3,996
Profit Per Month
3.63%
2.69%
6.08%
6.94%
7.38%
7.40%
6.89%
7.69%
13.07%
8.59%
14.32%
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Female
39.89%
768
Customers
Male
60.11%
Gender
Female
Male
1,722
5,630
242
3,666
City Tier
1
2
3
Retention Rate Trend
60.7%
59.9%
57.0%
56.5%
59.7%
57.9%
57.7%
53.7%
51.3%
62.8%
0
2
4
6
8
10
12
14
16
Quarters
Champion Products
i-Phone mini
Motorola Razr 2
HP - Laptop
i-Phone 14 Pro Max
i-Phone 13
Oppo F1 Selfie
Samsung M 21
One plus Nord
Samsung F5 Prime
Vaio Laptop
Redmi Note 5 Pro
Samsung S 22
Acer Laptop
Samsung Note 20
Samsung S22 Ultra
Vivo V5
Motorola Edge Plus
Nokia X 30 5G
i-mac
Mi Note 8 Pro
Dell Laptop
One Plus 10
LG G8X
Macbook Pro
i-Phone 14
Asus Computer
Asus Rog 3
Products Delivered Vs Distance Graph
8
295
389
444
559
274
233
221
249
550
288
322
217
89
75
74
85
104
96
85
86
89
81
71
81
94
101
94
67
63
93
51
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36

APPLIED CUSTOMER RETENTION STRATEGIES

- **Personalization:** Personalizing the customer experience by tailoring product recommendations, offers, and marketing messages to individual customers based on their preferences and behavior.
- **Excellent customer service:** Providing prompt and helpful customer support, resolving issues quickly and effectively, and showing appreciation for customer loyalty.
- **Rewards and loyalty programs**: Offering discounts, special offers, and other incentives to reward loyal customers and encourage repeat purchases.
- **Increasing customer engagement:** Building a sense of community among customers by creating social media groups, forums, or events where customers can connect and share their experiences. This also results in increasing customer engagement on website.
- **Continuous improvement**: Continuously improving the quality of products, services, and customer experiences based on customer feedback and market trends.
- **Quick delivery** : Advance supply chain, reduces delivery distance thus delivery time and cost.

(TABLEAU PRESENTATION)

CONCLUSION

- Customer retention is a crucial aspect of any business that can make the difference between success and failure. By focusing on retaining customers, businesses can reduce customer churn, improve brand loyalty, and drive long-term revenue growth. Throughout this project, we have conducted extensive research and analysis to understand the factors that contribute to customer retention and churn rates. Through this, we have identified various strategies that can be used to improve customer retention, such as personalized communication, excellent customer service, loyalty programs, and community building.
- By leveraging the insights and strategies developed in this project, businesses can take a proactive approach to customer retention and implement tailored strategies to meet their specific needs. Through the use of data analysis, businesses can better understand their customers and personalize their experiences, leading to increased satisfaction and loyalty. By adopting customer-centric strategies, businesses can build strong relationships with their customers, which can drive repeat business and positive word-of-mouth referrals.
- As the competitive landscape continues to evolve, businesses that prioritize customer retention will be better positioned to succeed. By investing in customer retention strategies, businesses can differentiate themselves from their competitors and build a loyal customer base that supports long-term growth and success.

VOTE OF THANKS

We would like to take this opportunity to extend our heartfelt thanks to everyone who has contributed to the successful completion of our group project on customer retention.

First and foremost, we would like to express our deepest gratitude to our project supervisors Ms. Nidhi Sharma and Mr. Mario Thokchom , whose unwavering support and guidance have been invaluable throughout the project. We are grateful for their expertise, constructive criticism, and encouragement that have helped us to stay focused and motivated.

We would also like to thank our whole batch who provided us with valuable insights and ideas during our discussions. Their suggestions and feedback were instrumental in shaping our research and analysis.

Finally, we would like to express our heartfelt appreciation to our families and friends who stood by us throughout the project, offering us encouragement and support during challenging times. Thank you all for your contributions to our project, which would not have been possible without your support and assistance.


Thank You