Maximize the customer retention by predicting the customer churn in retail

(Capstone Project – Group 8)

Bona fide record of work done by

Sreeshitha Sreedhar

Swaruba P

Tamilarasan G

Vandhana Priya V

Vijay Paudel B

Mentored by

Mr Jayveer Nanda

POST GRADUATE PROGRAM IN DATA SCIENCE AND ENGINEERING

August 2020

Great Lakes Institute of Management
Great Learning Institute

CHENNAI – 600 096

Table of Contents

ACKNOWLEDGEMENT

A project work is a job of great enormity and it cannot be accomplished by an individual without the help of experts. This work is a result of selfless help of many individuals. Our heartfelt thanks to our guide Mr. Jayveer Nanda for his inspiring guidance and support given to us throughout the course of this work. We would like to express our gratitude to Mr. YL Prasad, Mr. Regi Mathew, Mr. Jayanth Rasamsetti , Mr. Prince Kumar in helping us understand the concepts of Machine Learning, Mr. Mahesh Anand who lead us through the path of Statistics, Dr T. K. Senthil Kumar who taught us the ways of Exploring and inferring Data, Mr. Vinay Kumar Ganjagunte and Ms. Shambhavi Shukla for setting our foundation strong in Python, Mrs. Vidya Selvaraj, Mrs. Neha N and Ms. Namrata Thukral helping us perceive the art of extracting required data from the database. Our special thanks to the members of the committee, all the faculty and supporting staff of the Great Learnings for assisting us in this project. We are much indebted to all those who are contributing directly or indirectly to make this project a success.

Industry Review:

The retail industry consists of all companies who sell goods and services to consumers. There are many different retailers around the world, including grocery, convenience, discounts, independents, department stores, DIY, electrical and specialty. The retail industry shows a steady growth year-on-year and employs a huge number of workers worldwide. This is a highly competitive, fast-paced industry, which is vital to the economy.

In an increasingly competitive landscape, retail industry players must compete in a number of ways. These days, consumers want first-rate customer service and an integrated shopping experience. The rise in omni-channel retailing is confirmation of this. Consumers want to combine the benefits of traditional shopping with the convenience of using modern technology. Consumers may now be shopping online using their tablet or smartphone, or they could be on the high street in a bricks-and-mortar store. As such, retailers must now provide a hassle-free, seamless experience for the consumer in order to remain competitive.

Customer retention refers to a business's ability to attract and maintain repeat customers. You measure using your customer retention rate, which is the rate at which your business is able to retain those existing shoppers. 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%.

In retail, customer retention can help you understand not only how positive the customer experience is but also how you're able to meet customers' expectations. It goes beyond simply customer loyalty — retailers have to be able to fulfill the demands of returning customers too

The tea market in India is being driven by the healthy production and consumption of the beverage. In 2020, nearly 1.10 million tons of tea was consumed in the country. The Asia Pacific market is showcasing substantial growth owing to the increasing discretionary incomes among consumers in the region. The growing popularity of green tea and herbal tea to improve the beauty and health is primary fueling the market growth in the region.

SYNOPSIS:

The data taken for our project is one such dataset that belongs to online tea retail store, which is a historical data collected from 2008 – 2018. The data also consists of details about number of promotion mails sent to the customers. The main purpose of this project is to create a predictive machine learning model, using various supervised classification models and identifying the best model that can classify the churned customers from the dataset. This will in turn help to find the features which have higher impact on the customer retention.

Objectives:

1. To propose a predictive model that can assist in finding the customers who will be churned.

2. To study the effect of promotion emails sent to a customer-on-customer retention.

CHAPTER 1

1.1 INTRODUCTION:

1.1.1 Description of the Dataset:

This dataset can be used to understand what are the various marketing strategy based on consumer behavior that can be adopted to increase customer retention of a retail store.
An online tea retail store which sells tea of different flavors across various cities in India. The dataset contains data about the store's customers, their orders, quantity ordered, order frequency, city etc.

Features	Description
custid	Computer generated ID to identify customers throughout the database
retained	1, if customer is assumed to be active, 0 = otherwise
created	Date when the contact was created in the database - when the customer joined
firstorder	Date when the customer placed first order
lastorder	Date when the customer placed last order
esent	Number of emails sent
eopenrate	Number of emails opened divided by number of emails sent
eclickrate	Number of emails clicked divided by number of emails sent
avgorder	Average order size for the customer
ordfreq	Number of orders divided by customer tenure
paperless	1 if customer subscribed for paperless communication (only online)
refill	1 if customer subscribed for automatic refill
doorstep	1 if customer subscribed for doorstep delivery
favday	Customer's favorite delivery day
city	City where the customer resides in

1.1.2 Variable categorization:

Above mentioned features are classified into Categorical and Continuous as follows:

Categorical columns: custid, created, firstorder, lastorder, retained, paperless, refill, doorste p, favday, city

Numerical columns: esent, eopenrate, eclickrate, avgorder, ordfreq

1.2 Project Justification:

1.2.1 Project Statement:

Churn quantifies the number of customers who have unsubscribed or cancelled their service contract. Customers turning their back to your service or product are no fun for any business. It is very expensive to win them back once lost, not even thinking that they will not do the best word to mouth marketing if unsatisfied. So, we have to identify the customers who are not satisfied with the products/services and retain them.

The main objective domain is to study the effect of promotion emails on customer retention. With multiple competitors in the same business, it's really important to re-engage existing customers and keep them from churning.

1.2.2 Complexity involved:

The main expectation of a customer is hassle free traditional shopping that can be done in a modern way. To satisfy the customer expectation of shopping with one click, the sellers are adapting to modern way of selling their products online. Currently online retailing industry is booming because the customers expecting to receive their products at their doorstep. The biggest complexity involved in online retail industry is customers have so many options to switch on. The level of customer satisfaction for each customer depends on various factors (eg delivery time, packaging, combo offers, etc.) and the dependent factors for each customer is completely different because their needs are different. Since they have so many options to switch to other shopping sites, it is important for the retailer to retain their customers.

In order to retain the customers, the retailers adapted various marketing strategies and market basket analysis and various other methods to study their customer shopping pattern. As per our literature survey, we have seen that the customer churn was predicted mainly based on CRM records, feedbacks from social media, feedback survey conducted on behalf of the company. But our perspective is that most of the marketing is done through primary mode of communication which is email. So, in this project, we are going to analyze whether the promotional emails sent to the customer has any effect on customer churn.

1.2.3 Project Outcome:

Acquiring a new customer can cost five times more than retaining an existing customer.

Our predictive model predicts whether the customer is retained or not based on various factors like promotional emails sent, average order placed by the customer, order frequency, city etc. We will also be studying how the promotional emails aids our customer retention.

CHAPTER 2

2.1 Data Pre-Processing:

2.1.1 Null value imputation:

There were 6.4% of null values present in custid, created, firstorder and lastorder. As there are fewer null values, they are all dropped from the dataset.

2.1.2 Handling Non-Standard values and duplicated values:

From the above data descriptions, the columns created, lastorder, firstorder are supposed to be in datetime format. Before performing data type conversions non-standard missing values are handled. All these 12 non- standard missing values had been dropped from the dataset.

The range of years is supposed to be 2008 to 2018. But we have dates which are of year 1904. So, we had dropped the rows which has dates in year 1904.

There were some duplicated values in the column custid. 6 of these customers ids were repeated twice in the data and had dropped all these 12 rows due to inconsistent values.

There are some discrepancies in the date columns. Practically, Lastorder date should be greater than firstorder date and firstorder date should be greater than created date. So, the rows, which does not follow the above-mentioned rule, are dropped.

2.1.3 Alternate sources of data that can supplement the core dataset:

We have added 3 more columns that supplement the dataset. The columns are derived from dataset.

Tenure(int): The tenure of the customer calculated using difference between lastorder date and created date of the customer

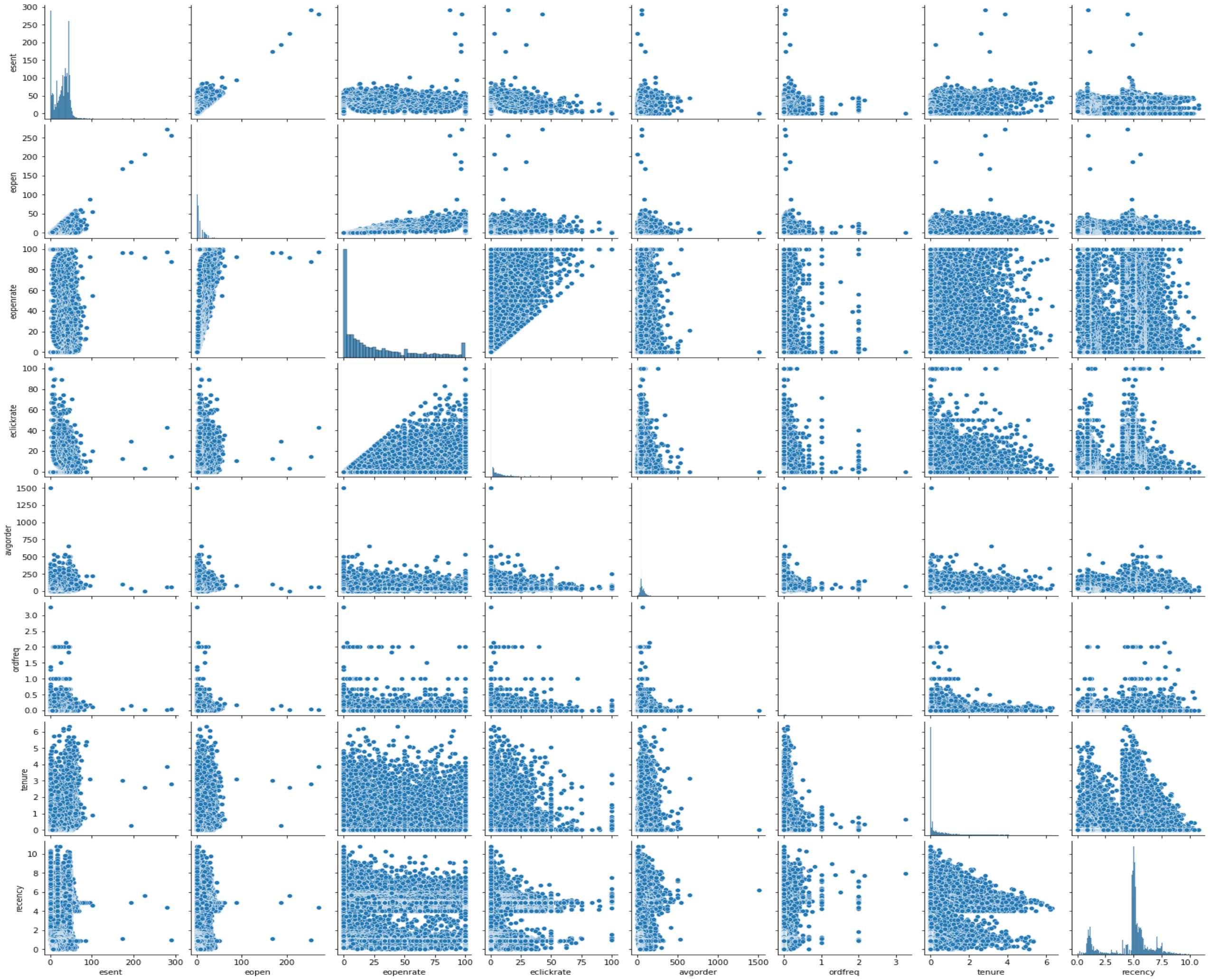
Recency (int): No. of days since the last order was placed by the customer (Difference between max_lastorder date and customer's lastorder date)

eopen(float): Number of emails opened by the customer

CHAPTER 3

3.1 EXPLORATORY DATA ANALYSIS:

3.1.1 Relationship between variables:

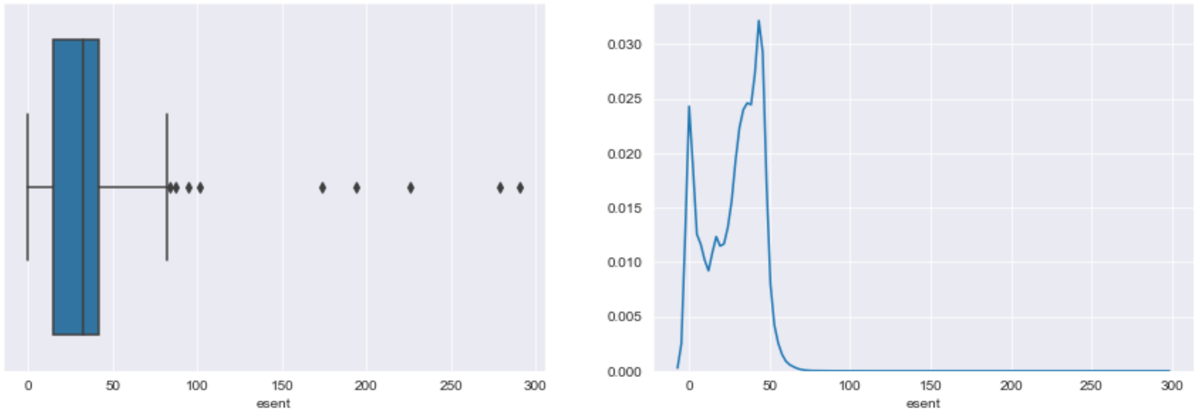


From the pair plots, we can see that esent and eopen has similar characteristics because the column eopen was derived from esent. Apart from Eopen and esent, other columns are not having linear relationship.

3.1.2 UNIVARIATE ANALYSIS:

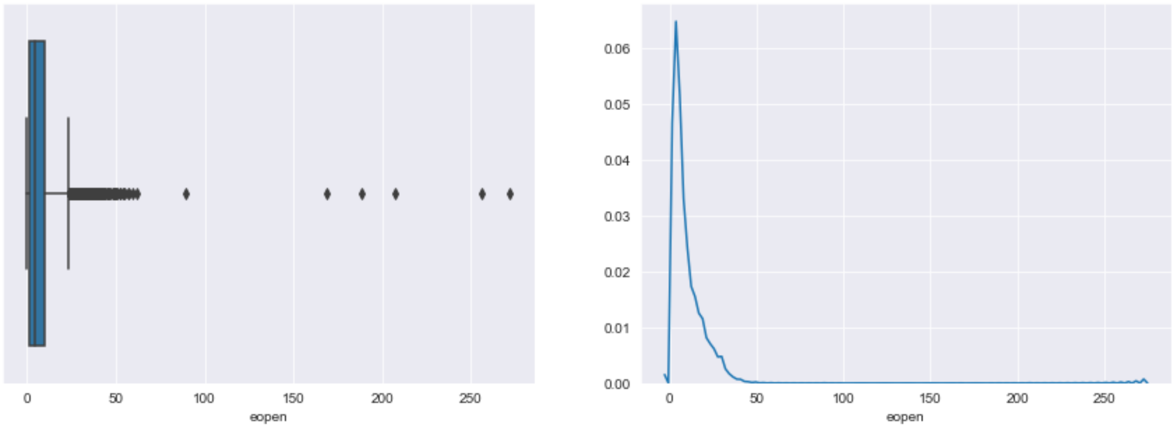
3.1.2.1 Numerical Columns:

1. **esent:**



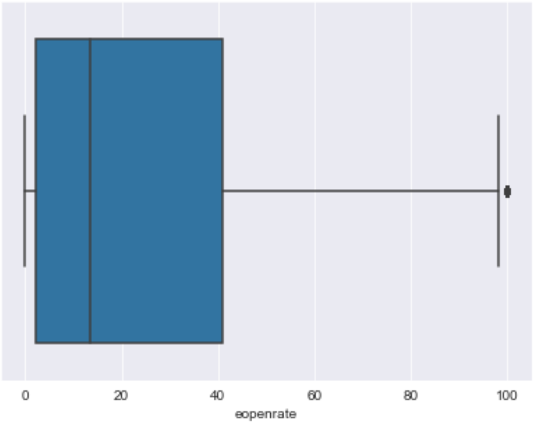
The above plot is the distribution of esent variable. From the descriptive analysis and skew value, it was shown that the data is right skewed and the outliers exist on positive side of the data. Whether these are extreme values or outliers, can be found based on domain knowledge. We can also see that most of the customers have received around 15-40 emails from the company.

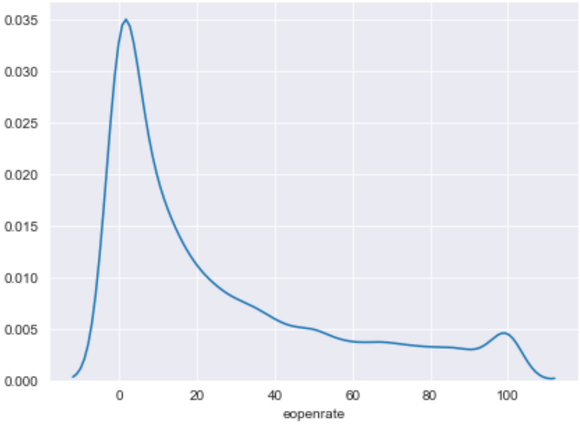
2. **eopen:**



The above plot is the distribution of eopen variable. From the descriptive analysis and skew value, we can see that the data is right skewed and the outliers exist on positive side of the data. The descriptive analysis shows that most of the customers have opened around 10 mails out of all promotional mails.

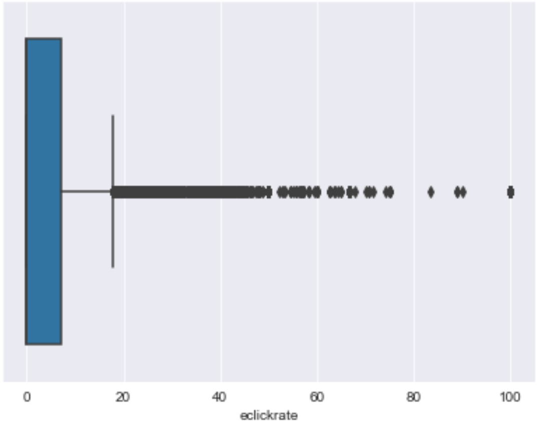
3. eopenrate:

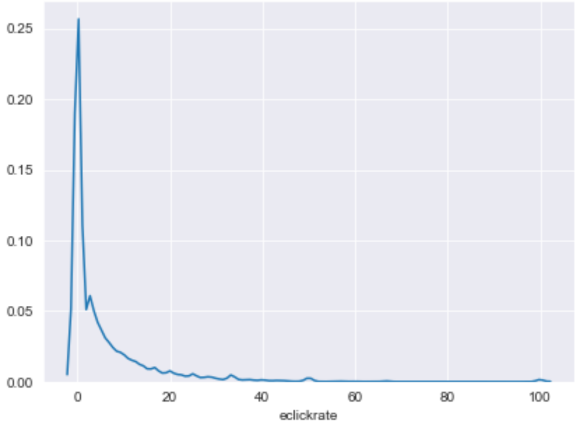




Since eopenrate gives the % measure of mails opened by customer with respect to mails sent to the customer, the range is 0 to 100 %. There is only 1 outlier which is at 100%. But this cannot be eliminated because, most of the emails opened was in the range 0-10. There is a chance that only one mail was sent to the customer and opened by the customer. So as per this point, it's not an outlier. Further analysis can be performed whether to check whether this value is an outlier or not.

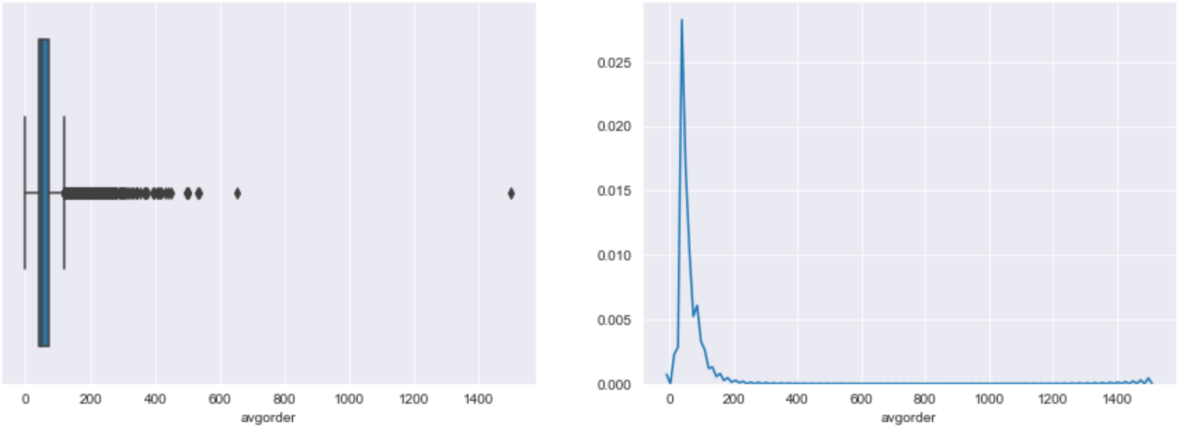
4. Eclickrate:





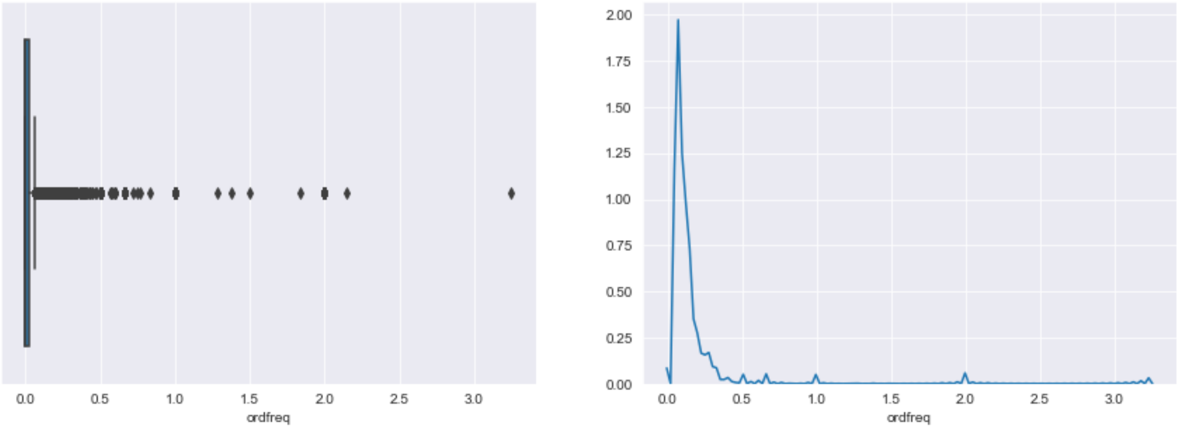
The above plot is the distribution of eclickrate variable. Eclickrate variable is a measure of number of links clicked in the email with respect to the number of emails received. From the descriptive analysis, it is seen that, 50% of the customers have never clicked the links in the email. They might have opened the mail, but they might not be interested in the offer or the product.

5. **Avgorder:**



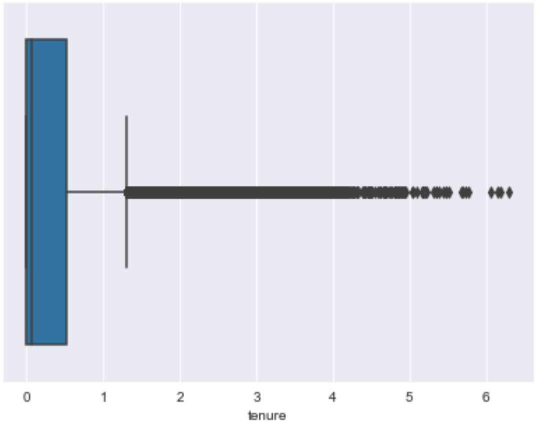
The above plot is the distribution of avgorder variable. This variable tells, on an average how much quantity of tea powder was ordered. From the descriptive analysis, it was seen that most of the customers have ordered around 40-70gms of tea on an average. The skewness shows there are outliers on the positive side.

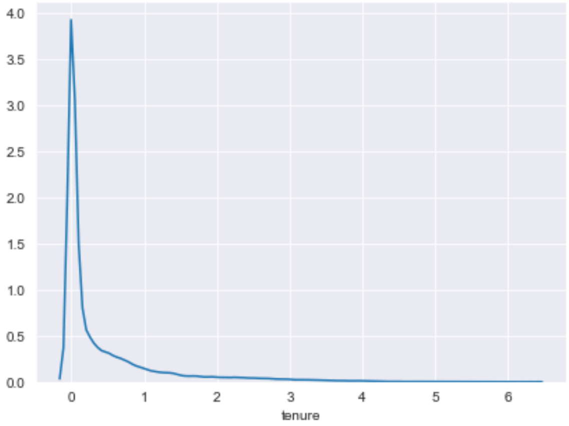
6. **Ordfreq**



The above plot is the distribution of ordfreq variable. Ordfreq tells how many orders customers have placed in this retail store with respect to their tenure. Descriptive analysis shows that, 75% of the customers have not placed the order again in the retail store. But since this is calculated with respect to tenure, there might be chance that the customer is retained for a long time, they have placed lesser number of orders. This can be understood with further bivariate analysis.

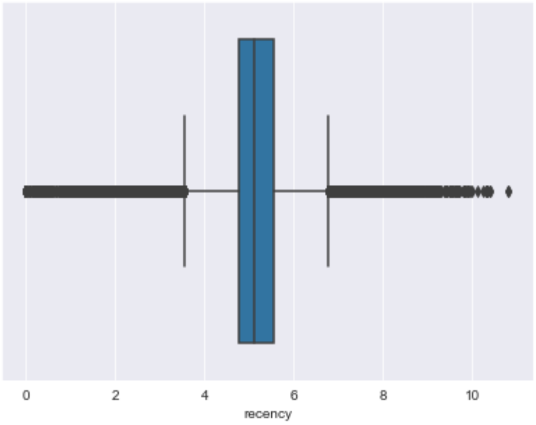
7. Tenure:

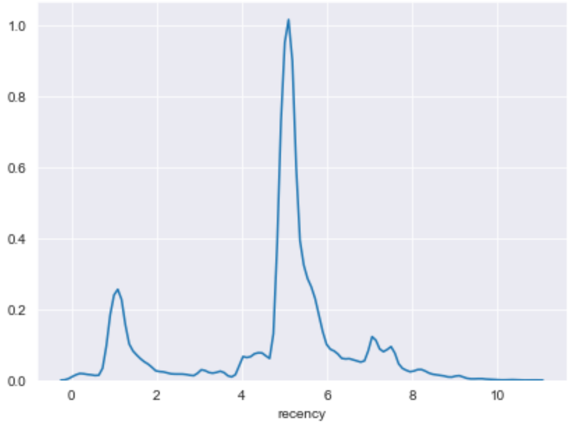




The above plot is the distribution of tenure variable. It tells how long the customer has been a member of/ordered from the retail store. Descriptive analysis shows that 75% of the customers have stayed for less than a year. Around 25% of the people have stayed more than a year.

8. Recency:

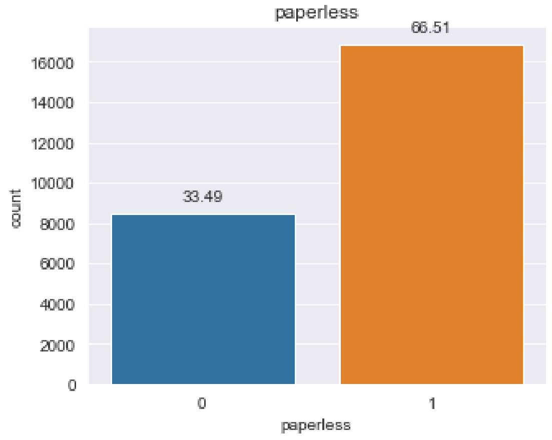


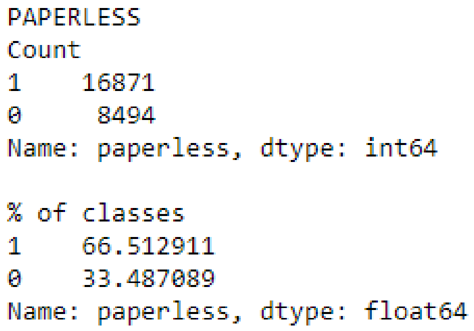


The above plot is the distribution of Recency variable. It tells how long it has been since the customer has placed the last order. The distribution is not skewed. The plot shows that most of the customer's recent orders were actually placed around 4 to 5 years back from the max lastorder date.

3.1.2.2 Categorical columns:

1. **Paperless:**

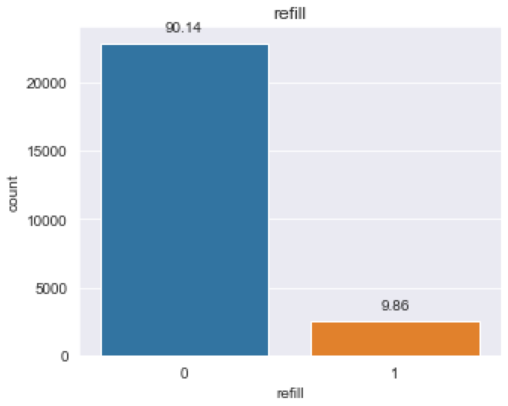


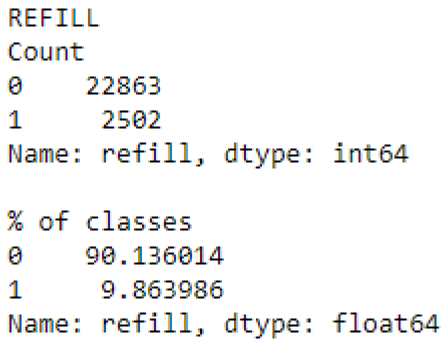


```
PAPERLESS
Count
1    16871
0     8494
Name: paperless, dtype: int64

% of classes
1    66.512911
0    33.487089
Name: paperless, dtype: float64
```

Paperless feature tells, if customers have opted for paperless communication. From the above countplot, it is seen that 67% of the cutomers have opted for paperless communication, whereas 33% of the customers have not opted for paperless communication.

2. **Refill:**

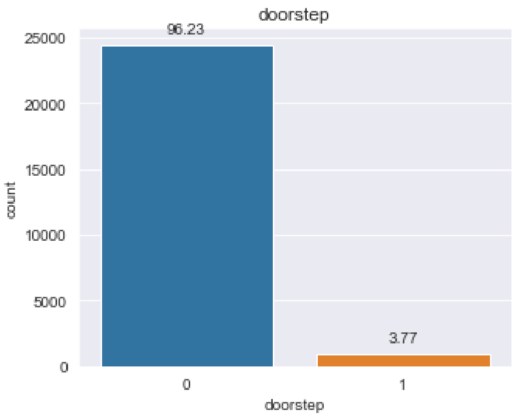


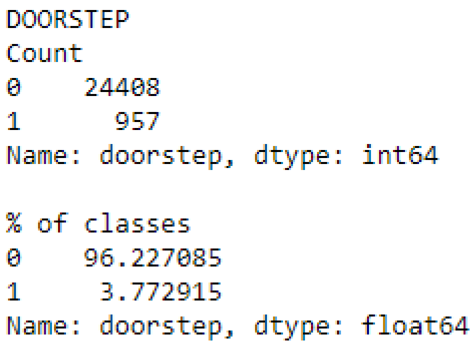


```
REFILL
Count
0    22863
1     2502
Name: refill, dtype: int64

% of classes
0    90.136014
1     9.863986
Name: refill, dtype: float64
```

Refill feature tells, if the customer wants the retail store to place the same order again every month automatically. From the above countplot, it is seen that 90% of the customers have not opted for a refill. Whereas 10% of the customers have opted for a refill.

3. Doorstep:

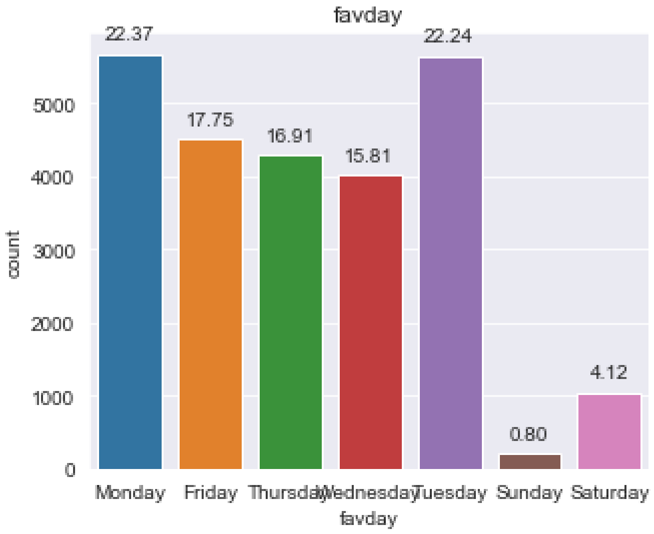


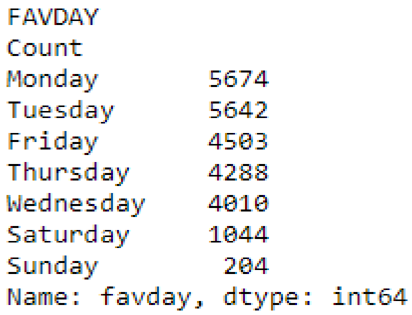


```
DOORSTEP
Count
0    24408
1      957
Name: doorstep, dtype: int64

% of classes
0    96.227085
1     3.772915
Name: doorstep, dtype: float64
```

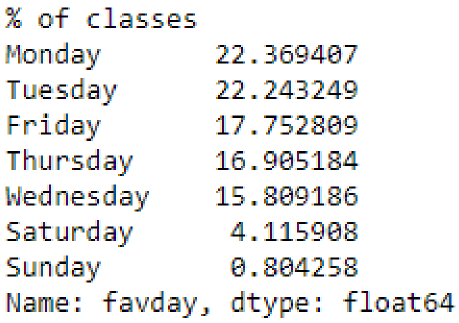
Doorstep feature tells, if the customer has opted for doorstep delivery. We can see that 96% of the customers did not opt for doorstep delivery. They wanted to pick up from the nearby store. Only 3% of the customers have opted for doorstep delivery.

4. Favday:





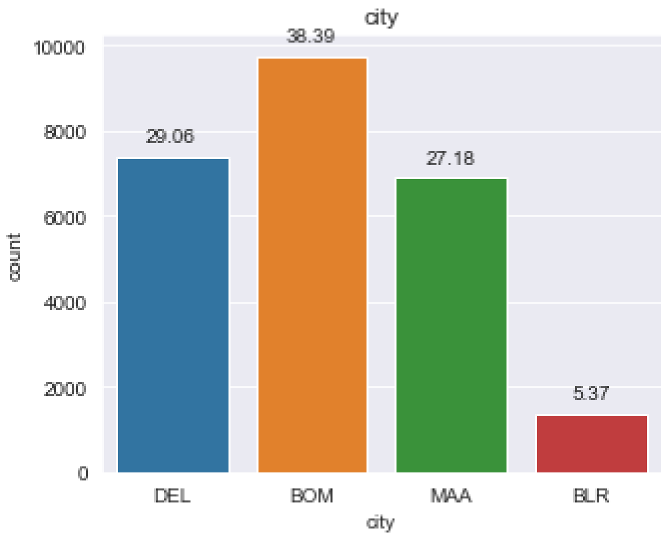
```
FAVDAY
Count
Monday       5674
Tuesday      5642
Friday       4503
Thursday     4288
Wednesday    4010
Saturday     1044
Sunday        204
Name: favday, dtype: int64
```

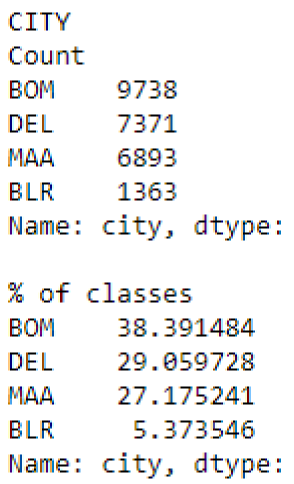


```
% of classes
Monday       22.369407
Tuesday      22.243249
Friday       17.752809
Thursday     16.905184
Wednesday    15.809186
Saturday      4.115908
Sunday        0.804258
Name: favday, dtype: float64
```

Favday variable tells, on which day customers have placed most of the orders. We can see that most of the customers have been placeing orders on Monday and Tuesday. Less number of orders were being placed on weekends.

5. City:

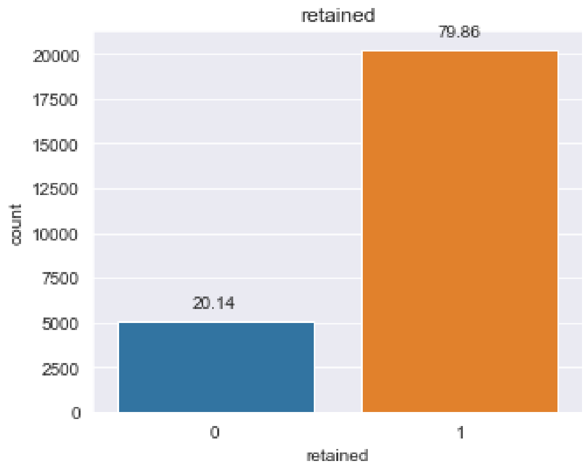


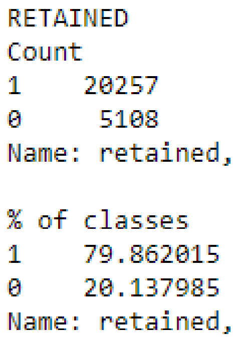


```
CITY
Count
BOM    9738
DEL    7371
MAA    6893
BLR    1363
Name: city, dtype:

% of classes
BOM    38.391484
DEL    29.059728
MAA    27.175241
BLR     5.373546
Name: city, dtype:
```

City variable contains information about city where the customer placed the order. Most of the orders were placed in Bombay and least number of orders were recorded in Bangalore.

6. Retained:





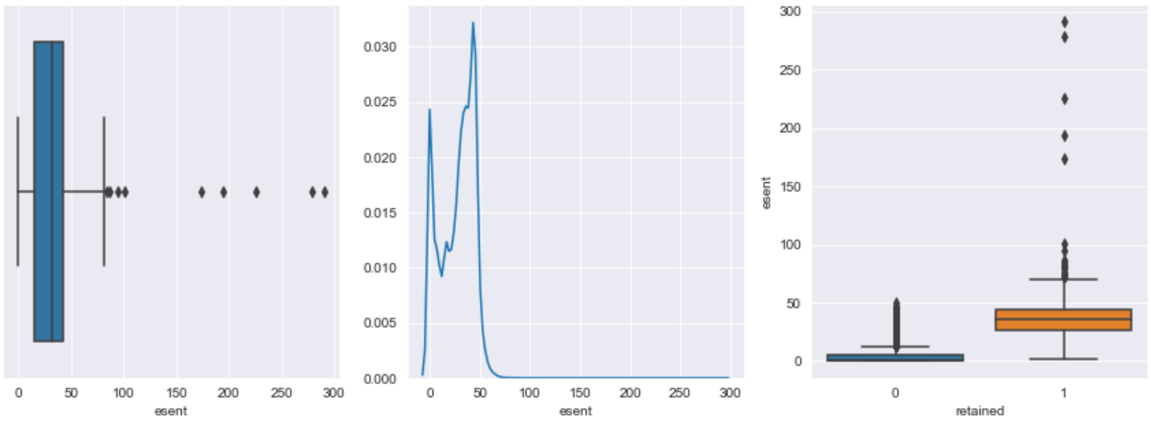
```
RETAINED
Count
1    20257
0     5108
Name: retained,

% of classes
1    79.862015
0    20.137985
Name: retained,
```

Retained variable is the target variable, which tells if the customer is retained or not. 1 represents the customer is retained and 0 represents that the customer is not retained, that is churned. The above countplot shows that 80% of the customers are retained and 20% of the customers are churned. This binary class variable shows that there is a class imbalance in the dataset.

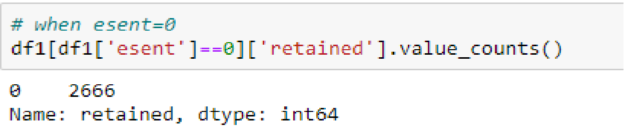
3.1.3 BIVARIATE ANALYSIS (Independent feature vs Dependent feature):

In the below bivariate analysis, each independent feature is compared with the target variable

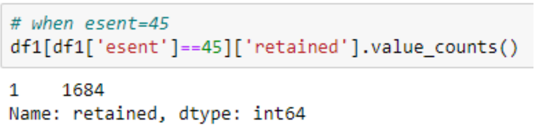
1. **Esent vs Retained:**



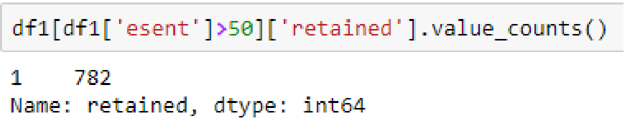
As we can see, there are few outliers in the esent column. The kdeplot shows there are two peaks - one near 0 and another one between 40-50. Also, above 50, kdeplot shows density values almost equal to 0. The range values for customer retained and not retained, shows a major difference. All the outlier values have a retained value of 1. Means higher the esent, the customer is more likely to be retained



```
# when esent=0
df1[df1['esent']==0]['retained'].value_counts()

0    2666
Name: retained, dtype: int64
```



```
# when esent=45
df1[df1['esent']==45]['retained'].value_counts()

1    1684
Name: retained, dtype: int64
```



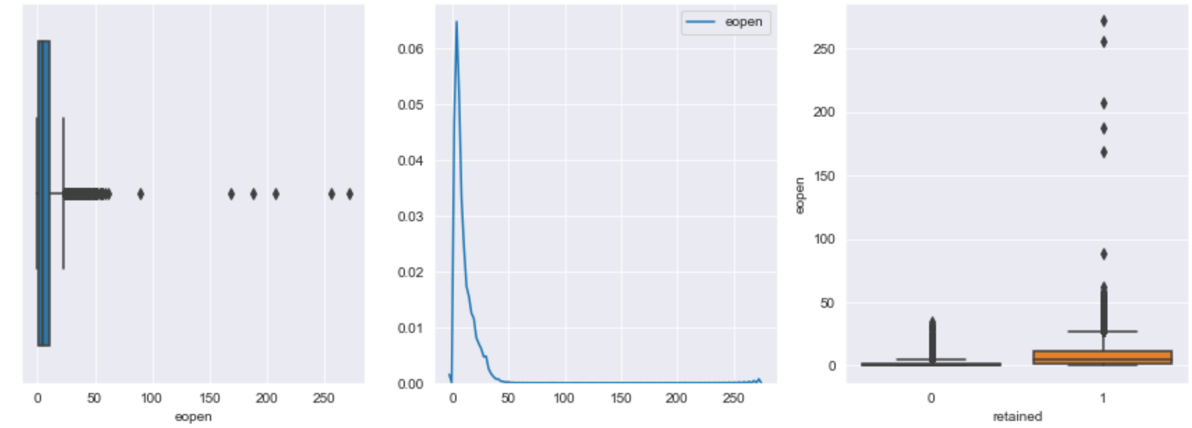
```
df1[df1['esent']>50]['retained'].value_counts()

1    782
Name: retained, dtype: int64
```

INSIGHTS:

Furthermore, analysis shows that, esent has a significance on the target variable 'retained'. Below are the observations:

- When no promotional mails were sent, none of the customers were retained.
- When more than 40 (in a range if 40 to 50) promotional mails were sent, customers were more likely to be retained.
- When more than 50 promotional mails were sent, all the customers were retained

2. Eopen vs Retained:



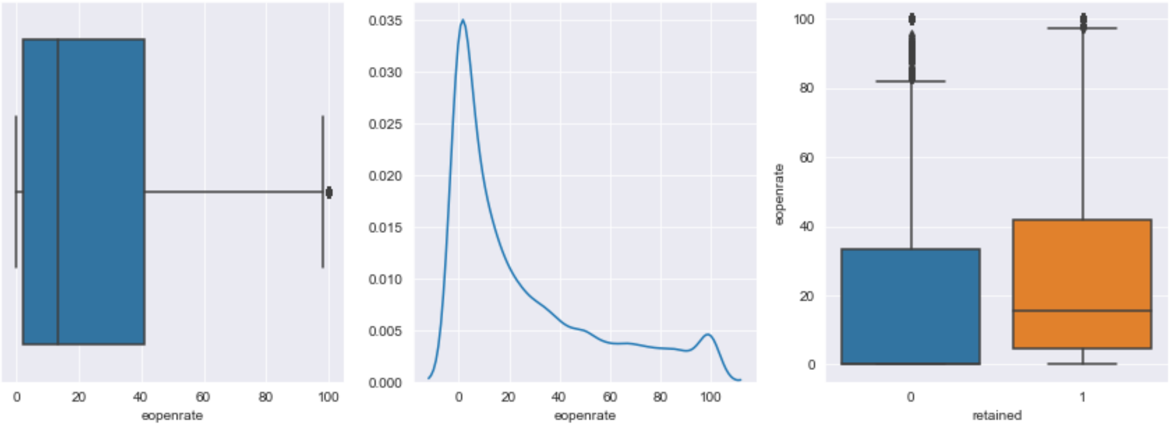
In eopen column, there are more outliers compared to esent. Since eopen was calculated based on esent column, there must be similarity of characters between them. Let's analyse eopen column using same modal analysis that we used for esent column. Boxplots between retained and eopen shows a significant difference. All the eopen outliers have retained=1

When none of the emails were opened by the customer, it does not show significant difference on target variable as esent had. Probability of retention of a customer who never opened a promotional mail from the retailer is same as that of the customer who will churn. (50-50 chance)

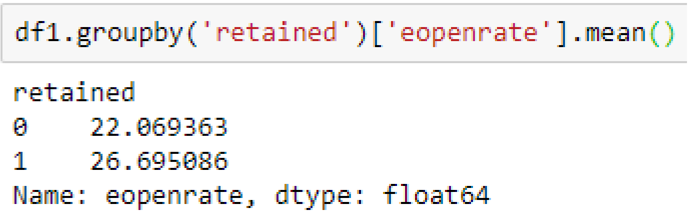
INSIGHTS:

- In this column, there are no outliers. They are all extreme values.
- When none of the emails were opened by the customer, it does not show significant difference on target variable as esent had. Probability of retention of a customer who never opened a promotional mail from the retailer is same as that of the customer who will churn. (50-50 chance)
- When eopen is greater than 40, the customer will definitely be retained. Because it makes sense that customer would open these many promotional mails only if the customer is interested in company's new launches, offers or any customized suggestions.
- When the customer has opened more than 9 promotional mails, there's a 97% possibility that the customer will be retained. Also, we can see that 50% the churned customers have never opened any of the promotional mails from retailer

3. Eopenrate vs Retained:



There is only 1 outlier in eopenrate which is equal to 100%. Actually, in our dataset we have 967 rows that have an eopenrate of 100%. There might be situations like, only 1 email was sent to the customer and the customer has opened that email, in that case eopenrate would be 100%. So, this cannot be considered an outlier but an extreme value which adds information to our model.



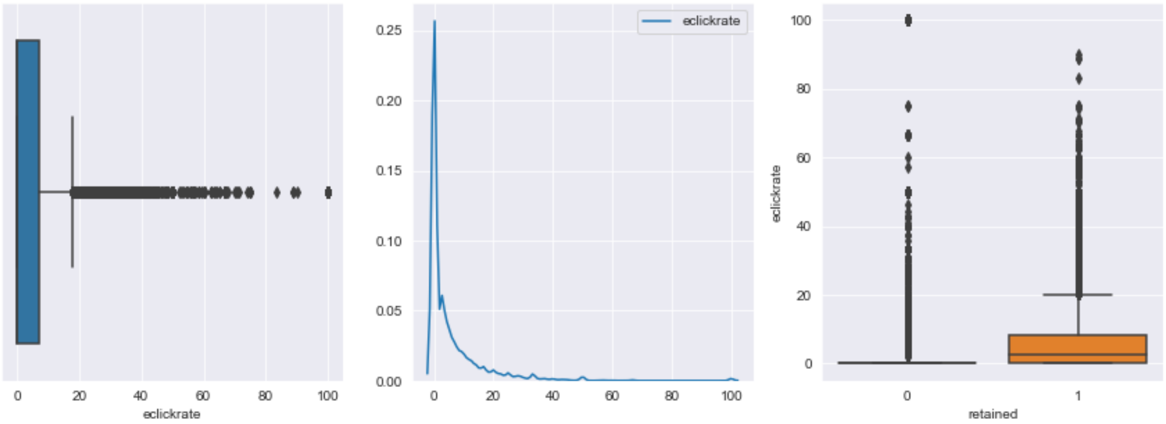
```
df1.groupby('retained')['eopenrate'].mean()
```

```
retained
0    22.069363
1    26.695086
Name: eopenrate, dtype: float64
```

INSIGHTS:

- There is only 1 outlier in eopenrate which is equal to 100%. Actually, in our dataset we have 967 rows that have an eopenrate of 100%. There might be situations like, only 1 email was sent to the customer and the customer has opened that email, in that case eopenrate would be 100%. So, this cannot be considered an outlier but an extreme value which adds information to our model
- Also, we can see that range of eopenrate for retained customers is slightly more than eopenrate range for churned customers. Though the mean value for both the groups differs slightly, it will add information to our model along with other features

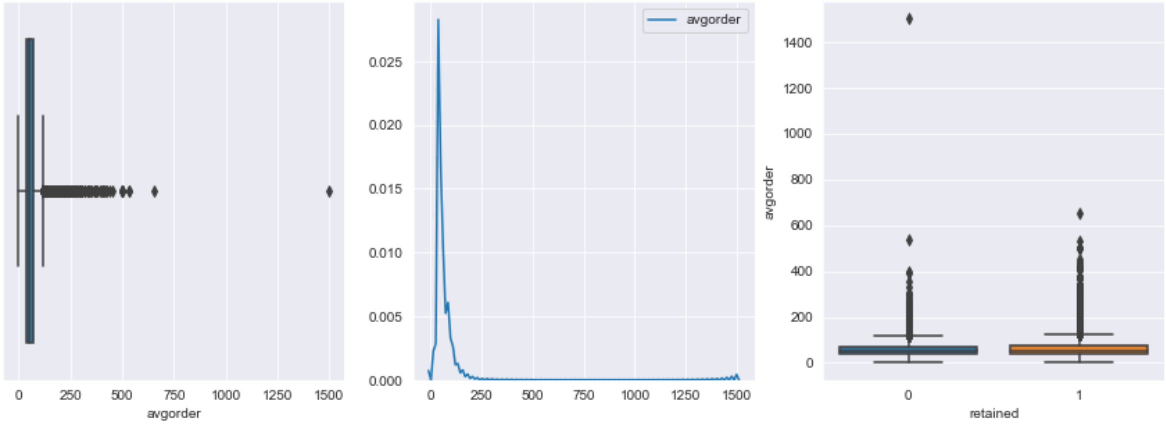
4. Eclickrate Vs Retained



INSIGHTS:

- From the descriptive analysis, we can see that, for 75% of the churned customers, eclickrate=0, which makes sense because either the customer hasn't opened the email or customer opened the mail but was not interested in the offers given.
- From the eopen data, we know that 50% of the churned customers have not opened any of the mails. So obviously their click rate would be zero. And so this pattern is seen in the churned customers. Since the averages vary slightly, along with other features, it can add information to our model

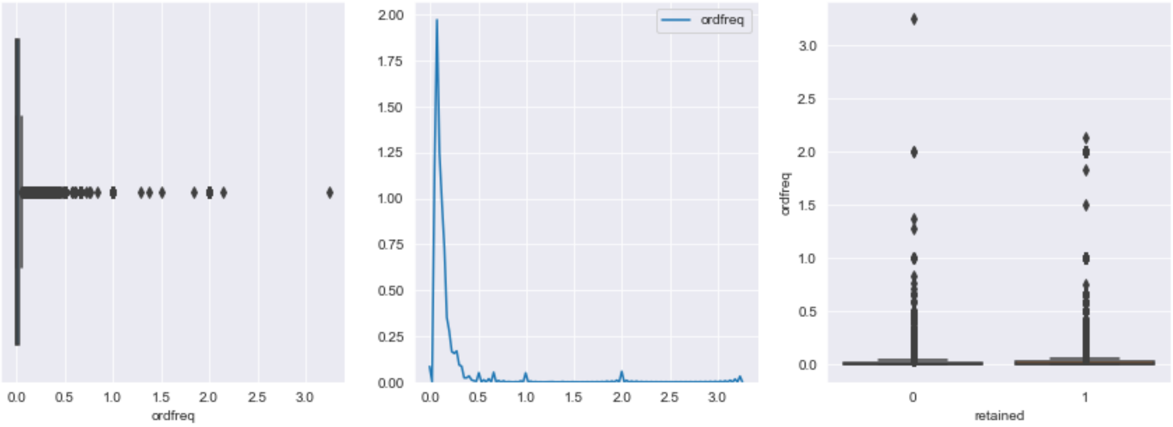
5. Avgorder vs Retained:



INSIGHTS:

- As per the domain knowledge, these values are not outliers. They are extreme values. The avgorder column shows no difference in the range values of retained and churned groups. Their averages are same. So, it does not add any information to our model. But still, we'll go with the statistical test to determine whether it is a significant feature or not.

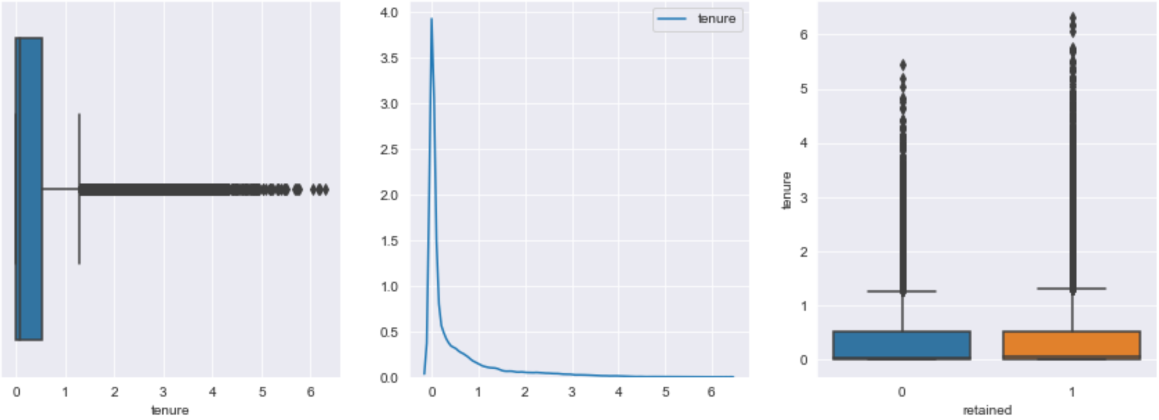
6. Ordfreq vs Retained:



INSIGHTS:

- As per the domain knowledge, these values are not outliers. They are extreme values.
- The ordfreq column shows no difference in the range values of retained and churned groups. Their averages are same. So, it does not add any information to our model. But still we'll go with the statistical test to determine whether it is a significant feature or not

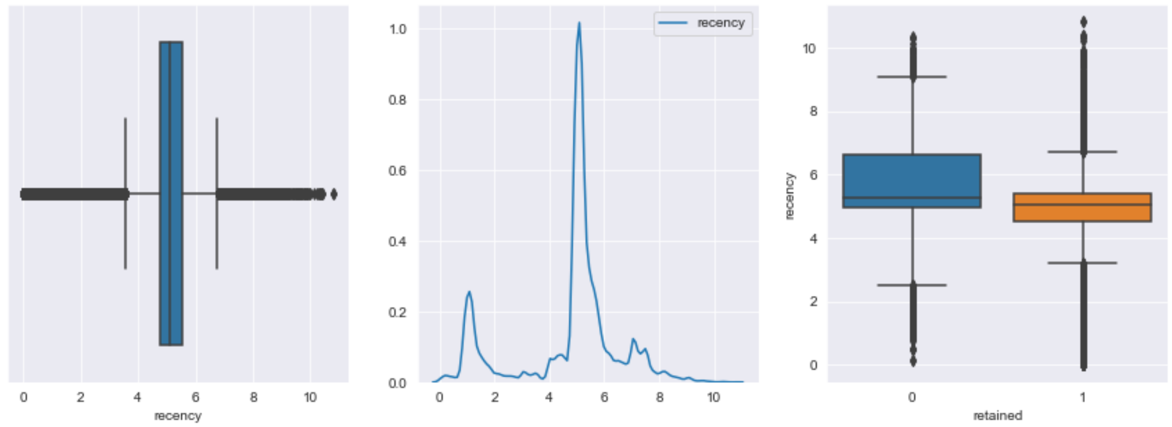
7. Tenure vs Retained:



INSIGHTS:

- Tenure column shows no difference in the range values of retained and churned groups. Their averages are same. So, it does not add any information to our model. But still we'll go with the statistical test to determine whether it's a significant feature or not
- Irrespective of the customer retention, around 75% of the customers in both the groups have shorter tenure of 0.5. As per the domain knowledge, these are extreme values.

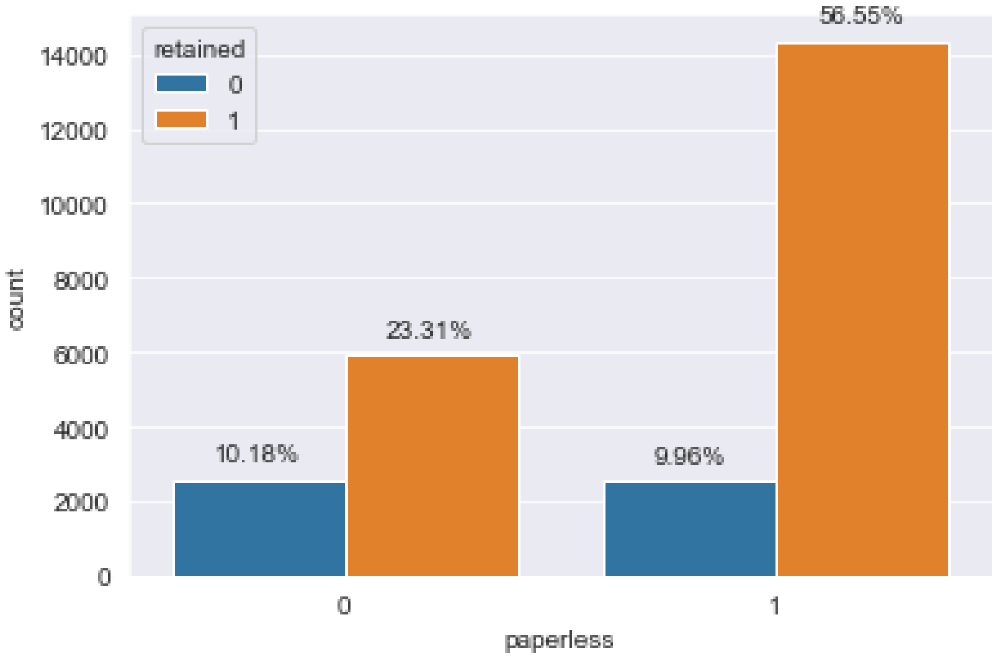
8. Recency vs Retained:



INSIGHTS:

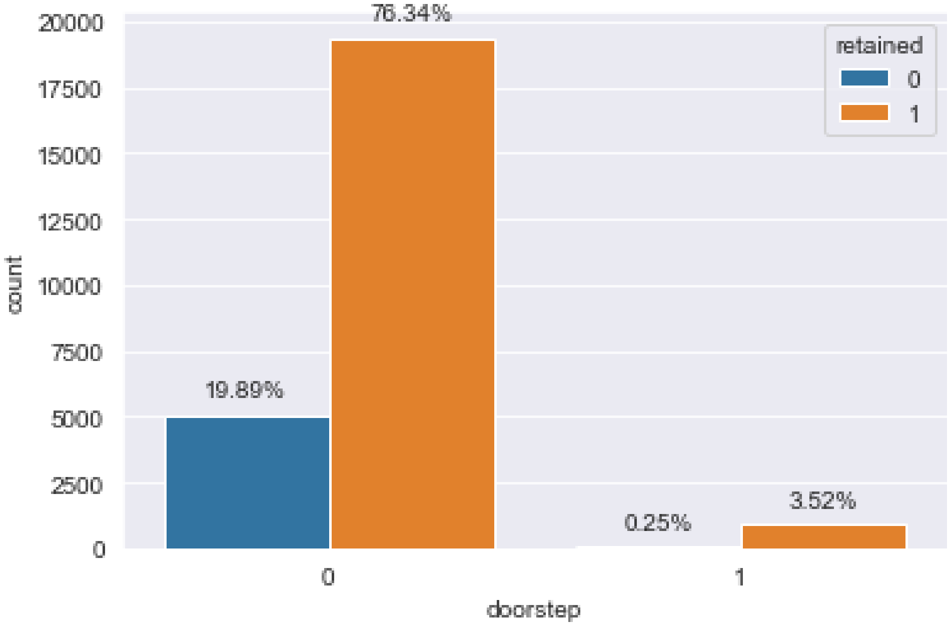
- As per the domain knowledge, these values are not outliers. They are extreme values.
- recency column shows no difference in the range values of retained and churned groups. Their averages are same. So, it does not add any information to our model. But still, we'll go with the statistical test to determine whether it's a significant feature or not

9. paperless vs retained:



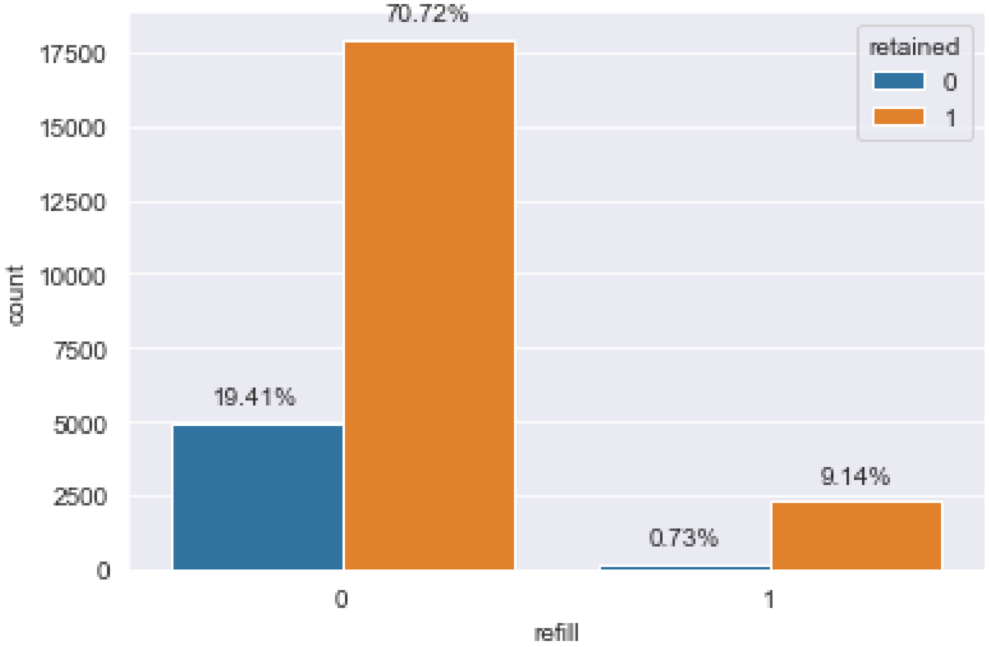
From the above comparison plot, we can see that, most of the customers who were retained, has opted for paperless communication. 10% of the customers who opted for paperless communication and 10% of the customers who did not opt for paperless communication, have been churned.

10. doorstep vs retained



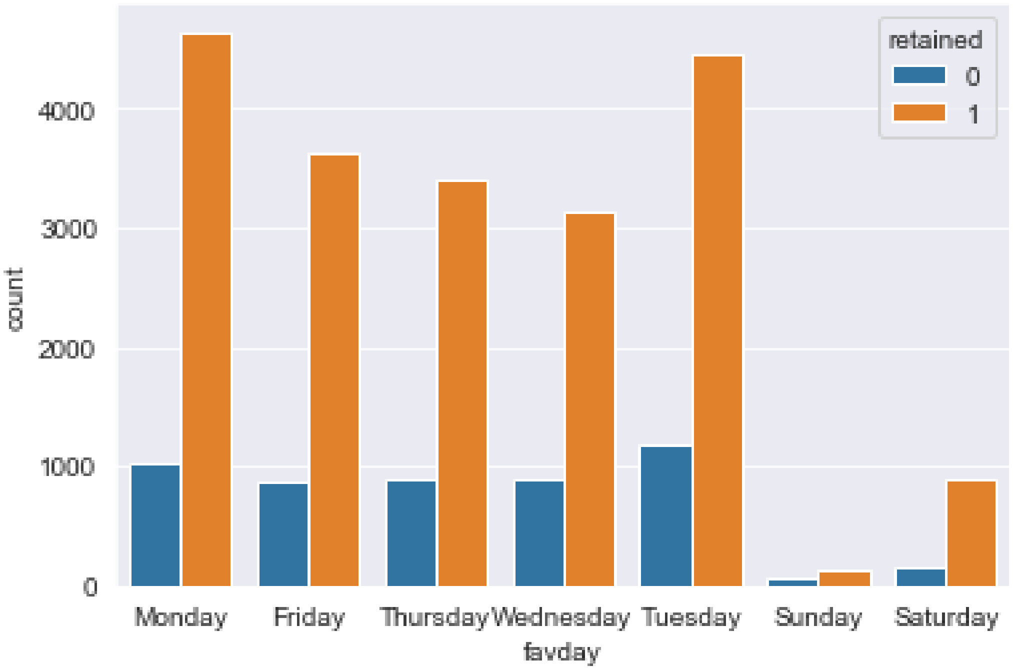
Less number of customers opted for doorstep delivery, and out of which only 3% where retained. Rest of the customers were churned. Most of the customers who did not opt for doorstep delivery has been retained, since our data has 80% of the customers have been retained.

11. refill vs retained



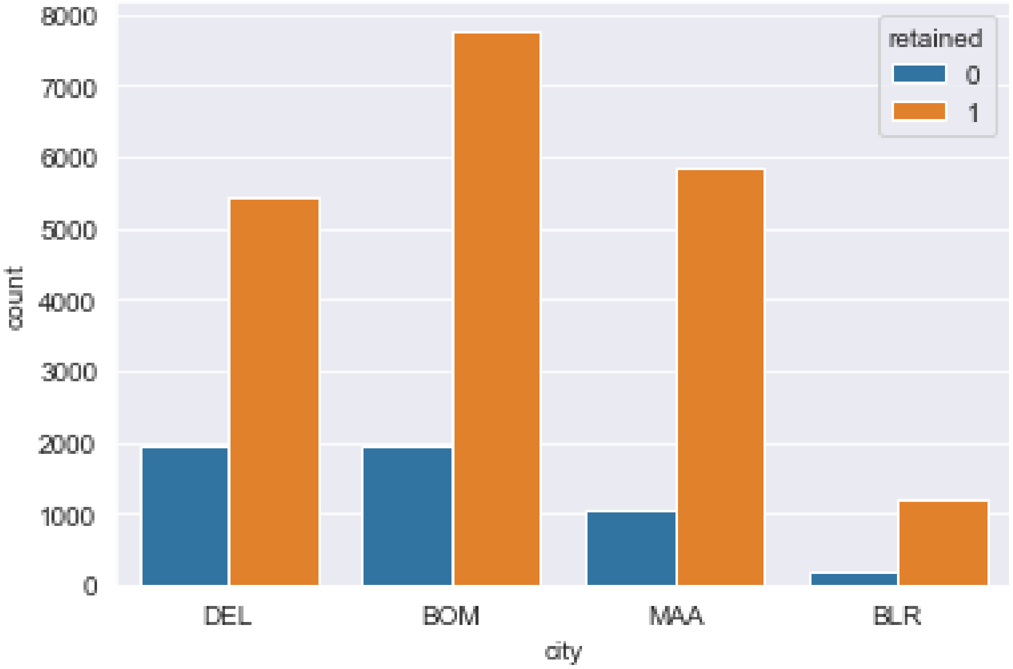
Very less number of people opted for Refill, and out of which most of them were retained.

12. favday vs retained



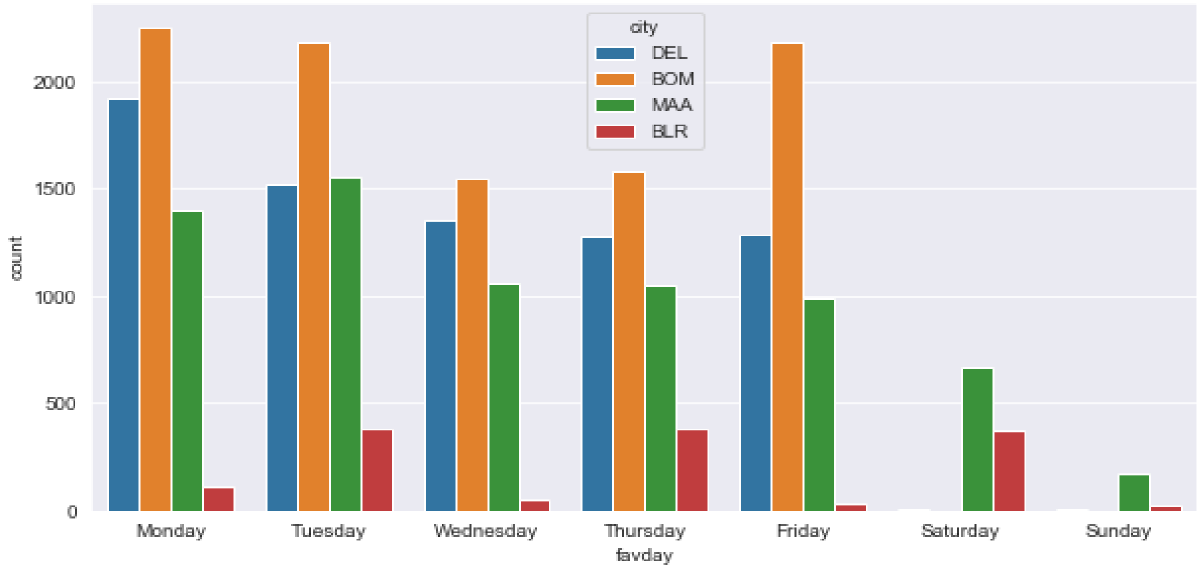
There is no much difference between the weekdays, but on weekends, we have very less customers placing the order. % of retained customers is same for all days.

13. city vs retained



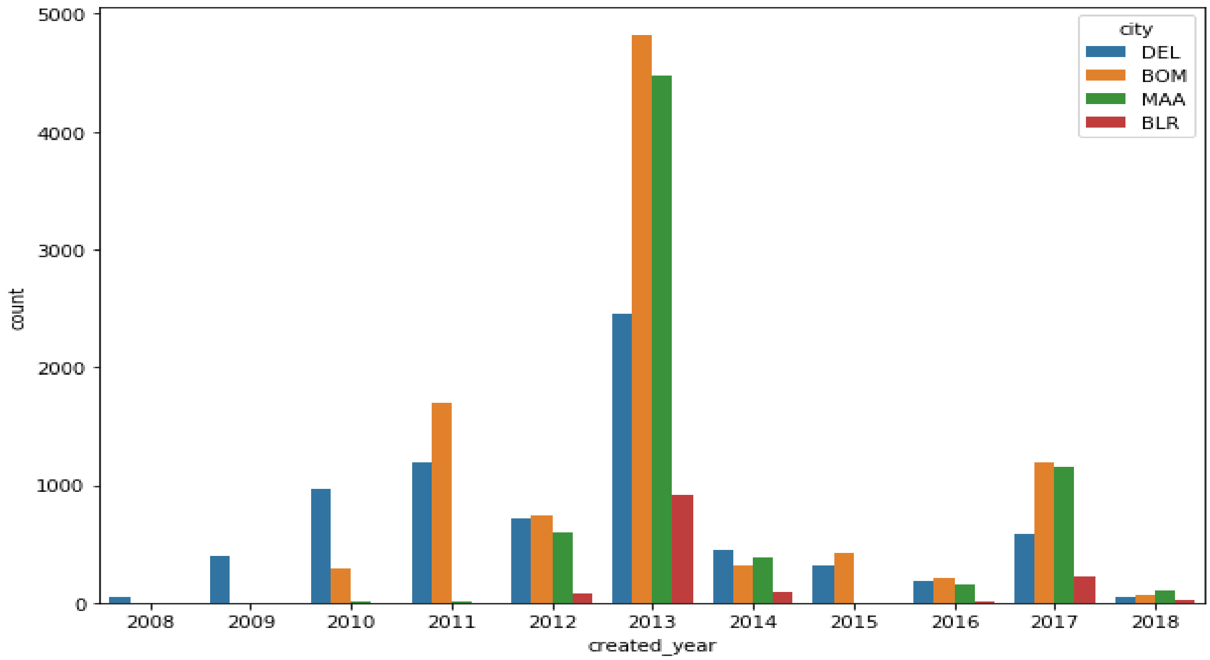
From the above plot, we can see that the distribution of retained and churned customers in each city has slight difference only except Bangalore. Because, a smaller number of customers were placed from Bangalore. Maximum number of customers were from Bombay and there were almost same number of customers from Delhi and Chennai.

14. favday vs city



From the above plot we can see that, no orders were placed on weekends in Delhi and Mumbai. On weekends, all the orders were placed by customers form Chennai and Delhi.

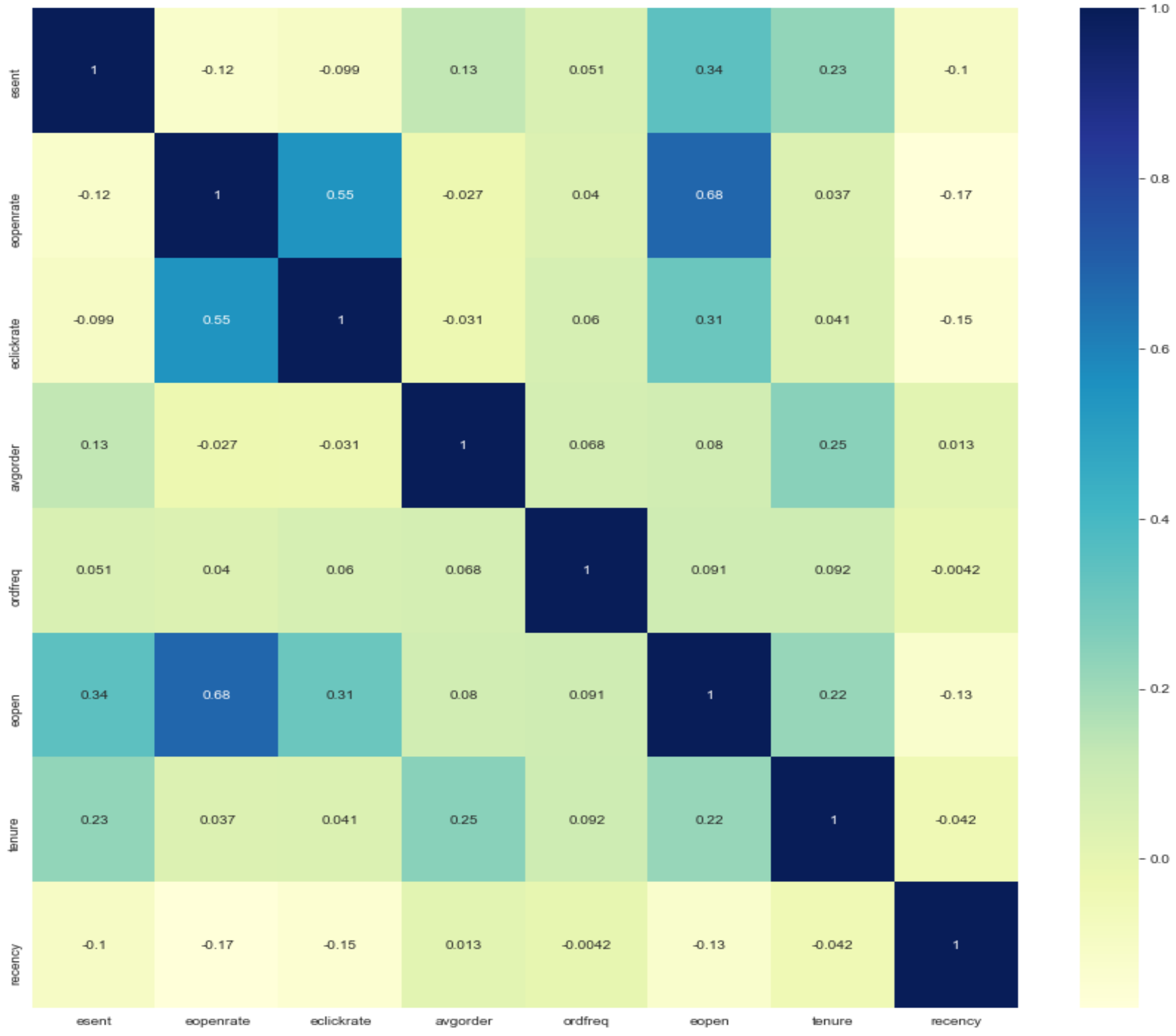
15. Created_year vs city:



- From the above plot, we can see that there were more new customers in 2013. In the year 2013, there is a spike in number of new customers who were interested in the retail store.
- In the years 2008 and 2009, we can see that there are no customers from Mumbai, Chennai and Bangalore. All the orders were placed from Delhi. This shows that the retail store was not launched in other cities until 2010.
- In 2010, we can see that some of the new customers were placing orders from Chennai.
- From 2012, we can see that some of the new customers were from Bangalore as well.
- But in 2015, there were no orders placed in Chennai and Bangalore.

3.1.4 Multivariate Analysis:

Multi-collinearity:



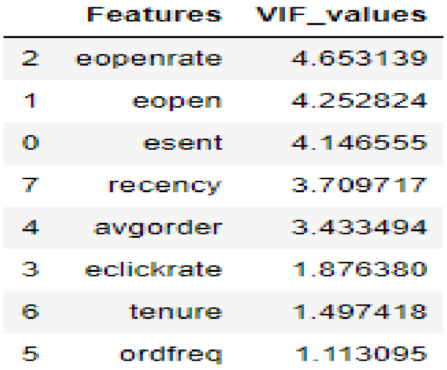
Collinear variables are features that have linear relationship with each other. The below details consist of features which are collinear with each other.

- esent is collinear with 2 features
- eopenrate is collinear with 3 features
- eclickrate is collinear with 2 features
- tenure is collinear with 2 features

Following table shows the features with high collinearity:

Feature 1	Feature 2	Collinearity
esent	Eopen	0.34
esent	tenure	0.23
eopenrate	Eclickrate	0.55
eopenrate	eopen	0.68
eclickrate	Eopenrate	0.55
eclickrate	eopen	0.31
Eopen	tenure	0.22

Further we have used the Variation Inflation Factor (VIF) from the statsmodels.stats.outliers_influence to select the features which reduces the collinearity within the features. It has returned the following features which results in the reduced collinearity between the features:



	Features	VIF_values
2	eopenrate	4.653139
1	eopen	4.252824
0	esent	4.146555
7	recency	3.709717
4	avgorder	3.433494
3	eclickrate	1.876380
6	tenure	1.497418
5	ordfreq	1.113095

By observing VIF values, eopenrate, eopen, esent has nearly equal to 4.5. So, we can drop the eopenrate column. Both eopen and eopenrate holds same information, only the measuring scale differs. In eopen, we count the number of emails opened by the consumer and in eopenrate, we calculate number of emails opened with respect to number of emails sent to the customer.

3.1.5 Statistical significance of variables:

Hypothesis Testing:

For shapiro Test:

H0 : Data is normal

Ha : Data is not normal

For numerical variables:

H0 : Independent numerical feature does not have any effect on target variable

Ha : Independent numerical feature has some effect on target variable

For categorical variables:

H0 : Independent categorical feature does not have any effect on target variable

Ha : Independent categorical feature has some effect on target variable

Level of Significance:

The level of significance for Shapiro, Mannwhitneyu, proportions_ztest, chi2_contingency = 5%

Declaration of Significance of a feature:

If the obtained p – value for each feature undergoing the respective statistical test is less than 0.05, then we reject the null hypothesis and therefore the feature is declared significant and can be used to predict the target variable.

If the obtained p – value for each feature undergoing the respective statistical test is greater than 0.05, then we fail to reject the null hypothesis and therefore the feature is declared non-significant and cannot be used to predict the target variable

Shapiro test for normality check:

Feature Name	P value	Inference
esent	0.0	The p-value is **lesser** than 0.05, so we **reject** the null hypothesis.
eopen	0.0	The p-value is **lesser** than 0.05, so we **reject** the null hypothesis.
eopenrate	0.0	The p-value is **lesser** than 0.05, so we **reject** the null hypothesis.
eclickrate	0.0	The p-value is **lesser** than 0.05, so we **reject** the null hypothesis.
avgorder	0.0	The p-value is **lesser** than 0.05, so we **reject** the null hypothesis.
ordfreq	0.0	The p-value is **lesser** than 0.05, so we **reject** the null hypothesis.
tenure	0.0	The p-value is **lesser** than 0.05, so we **reject** the null hypothesis.
recency	0.0	The p-value is **lesser** than 0.05, so we **reject** the null hypothesis.

Mannwhitneyu statistical test:

Feature Name	P value	Inference
esent	0.0	The p-value is **lesser** than 0.05, so we **reject** the null hypothesis.
eopen	0.0	The p-value is **lesser** than 0.05, so we **reject** the null hypothesis.
eopenrate	2.3859610254464484e-248	The p-value is **lesser** than 0.05, so we **reject** the null hypothesis.
eclickrate	0.0	The p-value is **lesser** than 0.05, so we **reject** the null hypothesis.
avgorder	0.37600770820703433	The p-value is **greater** than 0.05, so we **accept** the null hypothesis.
ordfreq	2.7892428883179996e-07	The p-value is **lesser** than 0.05, so we **reject** the null hypothesis.
tenure	0.05650878700623098	The p-value is **greater** than 0.05, so we **accept** the null hypothesis.
recency	5.237695566591219e-204	The p-value is **lesser** than 0.05, so we **reject** the null hypothesis.

Proportions_ztest:

Feature Name	P value	Inference
paperless	2.2378070185611648e-183	The p-value is **lesser** than 0.05, so we **reject** the null hypothesis.
refill	2.657417128645621e-63	The p-value is **lesser** than 0.05, so we **reject** the null hypothesis.
doorstep	1.5789618377712446e-26	The p-value is **lesser** than 0.05, so we **reject** the null hypothesis.

Chi2_contingency test:

Feature Name	P value	Inference
favday	3.214139452870288e-13	The p-value is **lesser** than 0.05, so we **reject** the null hypothesis.
city	5.308097236514491e-70	The p-value is **lesser** than 0.05, so we **reject** the null hypothesis.

3.1.6 Class imbalance and treatment:

The dataset consists of binary class, namely '1' and '0'. '1' denotes that the customer has retained and '0' represent that the customer doesn't retained.

In this dataset the Class '1' is of 24,425, which is of 79.46% and Class '0' is of 6,310, which is of 20.5%. There is an imbalance in the dataset, but this imbalance is the real-world scenario as the number of customers in class '0', tends to be very much less than the number of customers in class '1'. Therefore, it is better to train the model with this imbalance so that the model can learn this.

However, we will try the methods available to counter this class imbalance, such as SMOTE to understand how the model performance and report the same in the later part of the report.

CHAPTER 4

4.1 Feature Engineering:

4.1.1 Scaling the data:

To make it much easier for the model to learn the information from the data, we scale the data using the StandardScaler object from the Sklearn library.

The StandardScaler is first fit with the train data. This helps the StandardScaler object to learn the statistical data of the train data. Then we transform the train data with the StandardScaler object. Then with the same object the test data is also transformed, so that the test data is transformed with the same statistics of the train data.

4.1.2 Feature Selection:

The feature selection is used to select the statistically important features to predict the target variable. This usually helps in increasing the performance of the model by eliminating the less statistically important features. But it solely depends upon the nature of the dataset.

Here we have selected the significant features (all features except eopenrate, avgorder, tenure) based on the results from VIF and Statistical Tests.

CHAPTER 5

5.1 MODEL BUILDING:

The model building is done with the default parameters and then the hyper parameters of the model are tuned to refine the performance of the model to obtain the best possible result.

Classes in Target Variable:

0 - Customer not retained (Minority Class)

1 - Customer retained (Majority Class)

The model building follows the following steps:

- Base Model
- SMOTE
- Feature Selected using the Variation Inflation Factor and Statistical Tests
- Hyper Parameter Tuning

VIF and Statistical Tests:

Variation Inflation Factor (VIF) from the statsmodels.stats.outliers_influence, and based on statistical tests in order to select the features which reduces the collinearity within the features. Then using the selected features, the model is built.

SMOTE:

SMOTE from imblearn.over_sampling is used to improve the class imbalance in the target variable. The minority class is synthesized to represent certain percentage of the total dataset.With the obtained dataset the model is built. Here we have used the sampling strategy as 0.3, which in turn synthesizes the minority class to represent 20% of the total dataset.

Hyper Parameter Tuning:

The hyper parameter present in each of the model is tuned using Randomized Search Class and Grid Search class from sklearn.model_selection library. Then the hyper parameters combination is tested via grid search and Randomized Search, resulting in the hyper parameter that gives best score is chosen to build the model. This method is used across all the model building to verify the robustness of the result obtained.

Model Parameters:

There are various parameters to measure the performance of the classification model. Accuracy, ROC AUC Score, Precision, Recall, Sensitivity, Specificity is widely used one.

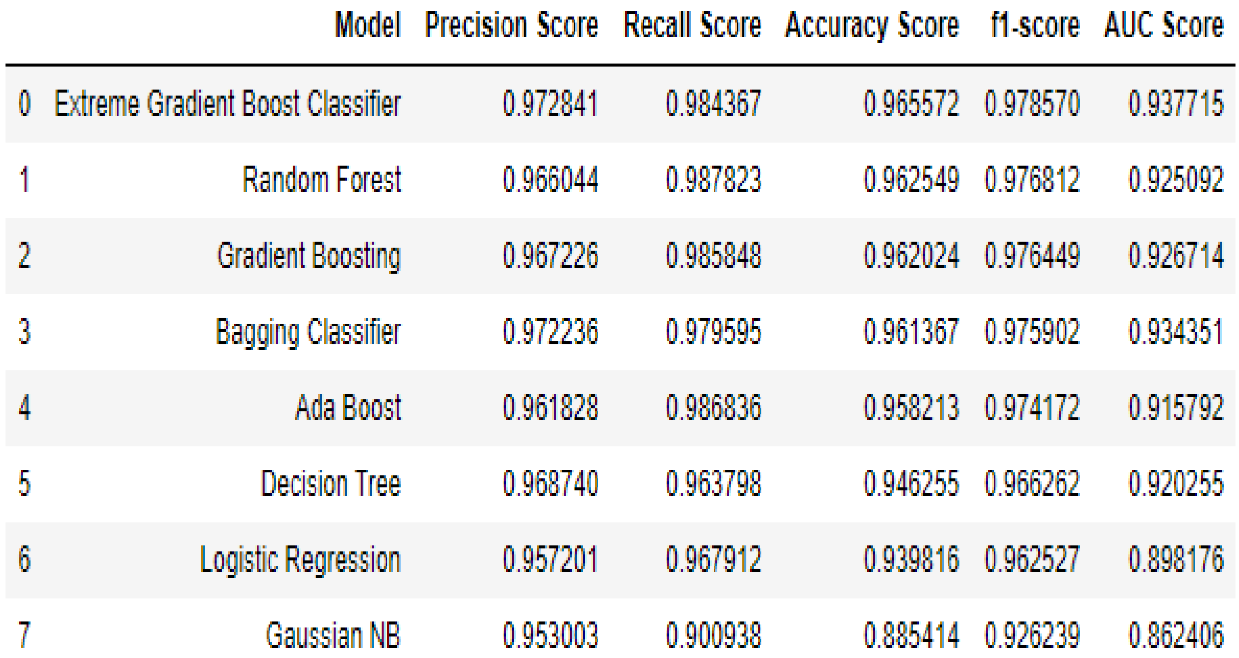
The performance of the classification models which uses the datasets with class imbalance are measured using the recall of the minority class. If the model has better recall value of the minority class, then that model is selected to be the best model. We have also built the model in line of obtaining the better recall score of the minority class and the model with better recall of minority class will be selected as the best model.

Evaluation Metrics chosen for our model is given below,

- F-measure has an intuitive meaning. It tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances).
  Unbalanced class, but both classes are important: If the class distribution is highly skewed (such as 80:20 or 90:10), then a classifier can get a low mis-classification rate simply by choosing the majority class. In such a situation, when we have imbalanced classes, the classifier that gets high F1 scores on both classes, as well as low mis-classification rate is chosen as the best classifier. A classifier that gets low F1-scores should be overlooked. F1 is maximum when precision = recall
- Accuracy is used when the True Positives and True negatives are more important while F1-score is used when the False Negatives and False Positives are crucial.

5.1.1 BASE MODEL:

We have done a basic model building and the performance of the models has been provided in the table below.



	Model	Precision Score	Recall Score	Accuracy Score	f1-score	AUC Score
0	Extreme Gradient Boost Classifier	0.972841	0.984367	0.965572	0.978570	0.937715
1	Random Forest	0.966044	0.987823	0.962549	0.976812	0.925092
2	Gradient Boosting	0.967226	0.985848	0.962024	0.976449	0.926714
3	Bagging Classifier	0.972236	0.979595	0.961367	0.975902	0.934351
4	Ada Boost	0.961828	0.986836	0.958213	0.974172	0.915792
5	Decision Tree	0.968740	0.963798	0.946255	0.966262	0.920255
6	Logistic Regression	0.957201	0.967912	0.939816	0.962527	0.898176
7	Gaussian NB	0.953003	0.900938	0.885414	0.926239	0.862406

Train and Test Metrics for Majority class:

Model	Precision		Recall		F1 Score	
	Train	Test	Train	Test	Train	Test
Logistic	0.96	0.96	0.97	0.97	0.96	0.96
Decision Tree	1.00	0.97	1.00	0.96	1.00	0.97
Random Forest	1.00	0.97	1.00	0.99	1.00	0.98
Ada boost	0.96	0.96	0.99	0.98	0.98	0.97
Gradient boost	0.97	0.97	0.99	0.98	0.98	0.98
Gaussian NB	0.96	1.00	0.90	1.00	0.93	1.00
Bagging	1.00	0.97	1.00	0.98	1.00	0.98
Extreme Gradient boost	0.99	0.97	1.00	0.98	0.99	0.98

Train and Test Metrics for Minority class:

Model	Precision		Recall		F1 Score	
	Train	Test	Train	Test	Train	Test
Logistic	0.88	0.86	0.82	0.83	0.85	0.85
Decision Tree	1.00	0.86	1.00	0.88	1.00	0.87
Random Forest	1.00	0.94	1.00	0.87	1.00	0.90
Ada boost	0.95	0.94	0.85	0.85	0.90	0.89
Gradient boost	0.95	0.94	0.88	0.87	0.91	0.90
Gaussian NB	0.68	0.68	0.85	0.84	0.75	0.75
Bagging	0.99	0.92	0.99	0.89	0.99	0.91
Extreme Gradient boost	0.99	0.94	0.97	0.89	0.98	0.91

Of all the models Ada boost, Extreme Gradient Boost, Gradient Boost and Random Forest comes in the top four based on the Recall value. As per our business problem both false positive and false negative affects the revenue of the retail store, so we have considered F1 score as our evaluation metric which gives the combined effect of precision and recall.

Based on the f1 score, Extreme Gradient Boost classifier, Random Forest, Gradient Boosting and Bagging classifier gives better results comparatively.

Above is the performance of the base models. Moving on, we will be doing the hyper parameter tuning for the applicable models and come to the conclusion of which model performs better in classifying the target variable.

5.1.2 SMOTE:

SMOTE is an oversampling technique where the synthetic samples are generated for the minority class. This algorithm helps to overcome the overfitting problem posed by random oversampling. Here SMOTE technique is used to treat the class imbalance in our data. It focuses on the feature space to generate new instances with the help of interpolation between the positive instances that lie together.

Sl.no	Model	Precision Score	Recall Score	Accuracy Score	f1-score	AUC Score
0	Extreme Gradient Boost Classifier	0.970848	0.984841	0.964518	0.977794	0.935700
1	Random Forest	0.967438	0.979726	0.973096	0.973544	0.973026
2	Bagging Classifier	0.968514	0.969223	0.968530	0.968868	0.968523
3	Gradient Boosting	0.951139	0.979482	0.964211	0.965102	0.964049
4	Decision Tree	0.961217	0.956522	0.958534	0.958864	0.958555
5	Ada Boost	0.947029	0.969468	0.957176	0.958117	0.957046
6	Gaussian NB	0.918404	0.882511	0.901024	0.900100	0.901221
7	Logistic Regression	0.903411	0.886419	0.894730	0.894834	0.894818

Train and Test Metrics for Majority class:

Model	Precision		Recall		F1 Score	
	Train	Test	Train	Test	Train	Test
Logistic	0.90	0.90	0.89	0.89	0.90	0.89
Decision Tree	1.00	0.96	1.00	0.96	1.00	0.96
Random Forest	1.00	0.97	1.00	0.98	1.00	0.97
Ada boost	0.95	0.95	0.97	0.97	0.96	0.96
Gradient boost	0.95	0.95	0.98	0.98	0.97	0.97
Gaussian NB	0.93	0.92	0.89	0.88	0.91	0.90
Bagging	1.00	0.97	1.00	0.97	1.00	0.97
Extreme Gradient boost	0.99	0.97	1.00	0.98	0.99	0.98

Train and Test Metrics for Minority class:

Model	Precision		Recall		F1 Score	
	Train	Test	Train	Test	Train	Test
Logistic	0.89	0.89	0.91	0.90	0.90	0.89
Decision Tree	1.00	0.95	1.00	0.96	1.00	0.96
Random Forest	1.00	0.98	1.00	0.97	1.00	0.97
Ada boost	0.97	0.97	0.95	0.94	0.96	0..96
Gradient boost	0.98	0.98	0.95	0.95	0.97	0.96
Gaussian NB	0.89	0.88	0.93	0.92	0.91	0.90
Bagging	1.00	0.97	1.00	0.97	1.00	0.97
Extreme Gradient boost	1.00	0.98	0.99	0.97	0.99	0.98

Here, the SMOTE model did not show much differences compared to the other models. There was a slight dip in the f1 scores compared to the base model.

5.1.3 Feature Selection based on VIF and Statistical tests:

We have used the Variation Inflation Factor (VIF) from the statsmodels.stats.outliers_influence, and based on statistical tests in order to select the features which reduces the collinearity within the features. We dropped 3 features (avgorder, tenure, eopenrate) based on VIF and statistical test.

Sl.No	Model	Precision Score	Recall Score	Accuracy Score	f1-score	AUC Score
0	Extreme Gradient Boost Classifier	0.970595	0.984344	0.963927	0.977421	0.934975
1	Bagging Classifier	0.973340	0.979871	0.962744	0.976594	0.938458
2	Random Forest	0.966139	0.985586	0.961167	0.975766	0.926540
3	Gradient Boosting	0.967261	0.983847	0.960773	0.975484	0.928053
4	Ada Boost	0.961026	0.986581	0.957619	0.973636	0.916551
5	Decision Tree	0.969046	0.964712	0.947566	0.966874	0.923252
6	Logistic Regression	0.958033	0.958748	0.933964	0.958390	0.898821
7	Gaussian NB	0.955913	0.899851	0.887640	0.927035	0.870326

Train and Test Metrics for Majority class:

Model	Precision		Recall		F1 Score	
	Train	Test	Train	Test	Train	Test
Logistic	0.96	0.96	0.97	0.96	0.96	0.96
Decision Tree	1.00	0.97	1.00	0.96	1.00	0.96
Random Forest	1.00	0.97	1.00	0.99	1.00	0.98
Ada boost	0.96	0.96	0.99	0.99	0.98	0.97
Gradient boost	0.97	0.97	0.99	0.98	0.98	0.98
Gaussian NB	0.96	0.96	0.90	0.90	0.93	0.93
Bagging	1.00	0.97	1.00	0.98	1.00	0.98
Extreme Gradient boost	0.99	0.97	1.00	0.98	0.99	0.98

Train and Test Metrics for Minority class:

Model	Precision		Recall		F1 Score	
	Train	Test	Train	Test	Train	Test
Logistic	0.87	0.85	0.83	0.83	0.85	0.84
Decision Tree	1.00	0.86	1.00	0.88	1.00	0.87
Random Forest	1.00	0.94	1.00	0.87	1.00	0.90
Ada boost	0.95	0.94	0.85	0.85	0.90	0.89
Gradient boost	0.95	0.93	0.88	0.87	0.91	0.90
Gaussian NB	0.68	0.69	0.85	0.84	0.75	0.76
Bagging	0.99	0.92	1.00	0.89	0.99	0.91
Extreme Gradient boost	0.99	0.94	0.96	0.89	0.97	0.91

The f1 score obtained for after the feature selection was comparatively similar to that of the base models.

5.1.4 Hyper Parameter Tuning:

1. LOGISTIC REGRESSION:

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression).

Hyperparameters:

Best Parameters: 'C': 0.1, 'penalty': 'l2'

Model	Precision Score	Recall Score	Accuracy Score	f1-score	AUC Score
Logistic	0.9563	0.9665	0.9381	0.9614	0.9658

Train and Test Metrics for Majority class:

Model	Precision		Recall		F1 Score	
	Train	Test	Train	Test	Train	Test
Logistic	0.96	0.96	0.96	0.97	0.96	0.96

Train and Test Metrics for Minority class:

Model	Precision		Recall		F1 Score	
	Train	Test	Train	Test	Train	Test
Logistic	0.86	0.86	0.83	0.83	0.84	0.84

Base model is built with all the features and it gives an f1 score of 0.96 in both the train and test datasets. From the above table we can see that, after selecting the significant features using VIF and statistical tests, we are getting the same scores as of the base model. By Applying GridSearchCV, for the logistic regression algorithm best parameters were found and passed to the final model. Even after Hyper Parameter tuning, the evaluation metrics are same that as before.

2. DECISION TREE:

A decision tree is a diagram or chart that helps determine a course of action or show a statistical probability. The chart is called a decision tree due to its resemblance to the namesake plant, usually outlined as an upright or a horizontal diagram that branches out. Starting from the decision itself (called a "node"), each "branch" of the decision tree represents a possible decision, outcome, or reaction. The furthest branches on the tree represent the end results of a certain decision pathway and are called the "leaves".

People use decision trees to clarify, map out, and find an answer to a complex problem. Decision trees are frequently employed in determining a course of action in finance, investing, or business. In mathematics, decision trees are also referred to as tree diagrams.

Hyper Parameters:

Decision Tree:

max_depth: 8
max_features: 9
min_samples_split: 2
criterion: gini

Bagged Decision Tree:

n_estimators: 10
max_features: 9

Ada Boost Decision Tree:

learning_rate: 0.015
n_estimators: 150

SL.No	Model	Precision Score	Recall Score	Accuracy Score	f1-score	AUC Score
0	Decision Tree	0.967846	0.985684	0.962418	0.976684	0.927936
1	Bagged Decision Tree	0.958778	0.991279	0.959001	0.974757	0.911164
2	Ada boost Decision Tree	0.938283	0.998190	0.946124	0.967310	0.868958

Train and Test Metrics for Majority class:

Model	Precision		Recall		F1 Score	
	Train	Test	Train	Test	Train	Test
Decision Tree	0.97	0.97	0.99	0.99	0.98	0.98
Bagged Decision Tree	0.99	0.96	1	0.99	1	0.97
Ada Boost Decision Tree	0.94	0.94	1	1	0.97	0.97

Train and Test Metrics for Minority class:

Model	Precision		Recall		F1 Score	
	Train	Test	Train	Test	Train	Test
Decision Tree	0.96	0.94	0.88	0.87	0.92	0.90
Bagged Decision Tree	1	0.96	0.96	0.83	0.98	0.89
Ada Boost Decision Tree	0.99	0.99	0.73	0.74	0.84	0.85

Base model is built with all the features and it gives an f1 score of 0.96 in both the train and test datasets. From the above table we can see that, after selecting the significant features using VIF and statistical tests, we are getting different scores from the base model. By Applying GridSearchCV, for the Decision Tree algorithm best parameters were found and passed to the final model. After Hyper Parameter tuning of Decision Tree, there was a slight increase in the evaluation metrics than before. Out of all the three models, tuned decision tree model gives the best f1 score of 0.97

3. RANDOM FOREST:

It is the method of constructing multiple decision trees on randomly selected data samples. We can use the bootstrap sampling method to select the random samples of the same size from the dataset to construct multiple trees. This method is used for both regression and classification analysis. The random forest returns the prediction based on all the individual decision trees prediction. For regression, it returns the average of all the predicted values; and for classification, it returns the class, which is the mode of all the predicted classes.

It avoids the over-fitting problem as it considers a random data sample to construct a decision tree. Random forest or random decision forest are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

Hyper Parameters:

{'n_estimators': 220, 'max_features': 'auto', 'criterion': 'entropy'}

Model	Precision Score	Recall Score	Accuracy Score	f1-score	AUC Score
Random Forest	0.968310	0.985519	0.962681	0.976839	0.928833

Train and Test Metrics for Majority class:

Model	Precision		Recall		F1 Score	
	Train	Test	Train	Test	Train	Test
Random Forest	1.00	0.97	1.00	0.99	1.00	0.98

Train and Test Metrics for Minority class:

Model	Precision		Recall		F1 Score	
	Train	Test	Train	Test	Train	Test
Random Forest	1.00	0.94	1.00	0.86	1.00	0.90

Base model is built with all the features and it gives an f1 score of 0.9768 in both the train and test datasets. From the above table we can see that, after selecting the significant features using VIF and statistical tests, we are getting same scores as that of the base model. By Applying RandomizedSearchCV, for the random forest algorithm best parameters were found and passed to the final model. Even After Hyper Parameter tuning of random forest, the evaluation metrics were same as that of base model.

4. NAIVE BAYES ALGORITHM:

Naïve Bayes is a probabilistic machine learning algorithm used for many classification functions and is based on the Bayes theorem. Gaussian Naïve Bayes is the extension of naïve Bayes. While other functions are used to estimate data distribution, Gaussian or normal distribution is the simplest to implement as you will need to calculate the mean and standard deviation for the training data

Hyperparameters:

'var_smoothing': 1.0

Model	Precision Score	Recall Score	Accuracy Score	f1-score	AUC Score
Naïve Bayes Algorithm	0.9408	0.934	0.9408	0.9374	0.8506
Bagged Naïve Bayes Algorithm	0.9426	0.9319	0.9003	0.9372	0.8534

Train and Test Metrics for Majority class:

Model	Precision		Recall		F1 Score	
	Train	Test	Train	Test	Train	Test
Naïve Bayes Algorithm	0.94	0.94	0.93	0.93	0.94	0.94
Bagged Naïve Bayes Algorithm	0.94	0.94	0.93	0.93	0.94	0.94

Train and Test Metrics for Minority class:

Model	Precision		Recall		F1 Score	
	Train	Test	Train	Test	Train	Test
Naïve Bayes Algorithm	0.74	0.75	0.78	0.77	0.76	0.76
Bagged Naïve Bayes Algorithm	0.74	0.74	0.78	0.78	0.76	0.76

Base model is built with all the features and it gives an f1 score of 0.926 in both the train and test datasets. From the above table we can see that, after selecting the significant features using VIF and statistical tests, we are getting different scores from the base model. By Applying GridSearchCV, for the Naïve Bayes algorithm best parameters were found and passed to the final model. After Hyper Parameter tuning of Naïve Bayes, there was a slight increase in the evaluation metrics than before. Out of all the three models, tuned Naïve Bayes model gives the best f1 score of 0.937.

5. GRADIENT BOOSTING:

GB builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage n_classes_ regression trees are fit on the negative gradient of the binomial or multinomial deviance loss function. Binary classification is a special case where only a single regression tree is induced.

This method optimizes the differentiable loss function by building the number of weak learners (decision trees) sequentially. It considers the residuals from the previous model and fits the next model to the residuals. The algorithm uses a gradient descent method to minimize the error.

Hyperparameters:

{'learning_rate': 0.1, 'n_estimators': 750}

Model	Precision Score	Recall Score	Accuracy Score	f1-score	AUC Score
Gradient boost	0.9706	0.9838	0.9633	0.9772	0.9812

Train and Test Metrics for Majority class:

Model	Precision		Recall		F1 Score	
	Train	Test	Train	Test	Train	Test
Gradient boost	0.97	0.97	0.99	0.98	0.98	0.98

Train and Test Metrics for Minority class:

Model	Precision		Recall		F1 Score	
	Train	Test	Train	Test	Train	Test
Gradient boost	0.95	0.93	0.87	0.88	0.91	0.91

Base model is built with all the features and it gives an f1 score of 0.976 in both the train and test datasets. From the above table we can see that, after selecting the significant features using VIF and statistical tests, we are getting same scores as the base model. By Applying GridSearchCV, for the gradient boosting algorithm best parameters were found and passed to the final model. Even after Hyper Parameter tuning of gradient boosting algorithm, the evaluation metrics obtained was same as before.

6. XGBOOST:

XGBoost (extreme gradient boost) is an alternative form of gradient boosting method. This method generally considers the initial prediction as 0.5 and build the decision tree to predict the residuals. XGBoost applies a better regularization technique to reduce overfitting, and it is one of the differences from the gradient boosting.

Hyperparameters:

```
{'subsample': 0.8,
 'n_estimators': 60,
 'min_samples_split': 4600,
 'min_child_weight': 5,
 'max_features': 16,
 'max_depth': 25,
 'learning_rate': 0.1,
 'gamma': 3,
 'colsample_bytree': 0.6}
```

Model	Precision Score	Recall Score	Accuracy Score	f1-score	AUC Score
Extreme Gradient Boost Classifier	0.968811	0.986506	0.963863	0.977578	0.930305

Train and Test Metrics for Majority class:

Model	Precision		Recall		F1 Score	
	Train	Test	Train	Test	Train	Test
Extreme Gradient boost	0.98	0.97	0.99	0.99	0.98	0.98

Train and Test Metrics for Minority class:

Model	Precision		Recall		F1 Score	
	Train	Test	Train	Test	Train	Test
Extreme Gradient boost	0.96	0.94	0.90	0.88	0.93	0.91

Base model is built with all the features and it gives an f1 score of 0.978 in both the train and test datasets. From the above table we can see that, after selecting the significant features using VIF and statistical tests, we are getting same scores as the base model. By Applying RandomizedSearchCV, for the extreme gradient boosting algorithm best parameters were found and passed to the final model. Even after Hyper Parameter tuning of extreme gradient boosting algorithm, the evaluation metrics obtained was same as before.

5.1.5 Best Model:

Comparing all the model results, the following have given best results:

Sl.No	Model	Precision Score	Recall Score	Accuracy Score	f1-score	AUC Score
0	Extreme Gradient Boost Classifier	0.9688	0.9865	0.9638	0.9775	0.9303
1	Gradient Boost	0.9706	0.9838	0.9633	0.9772	0.9812
2	Random Forest	0.9683	0.9855	0.9626	0.9768	0.9288

Train and Test Metrics for Majority class:

Model	Precision		Recall		F1 Score	
	Train	Test	Train	Test	Train	Test
Extreme Gradient boost	0.98	0.97	0.99	0.99	0.98	0.98
Gradient boost	0.97	0.97	0.99	0.98	0.98	0.98
Random Forest	1.00	0.97	1.00	0.99	1.00	0.98

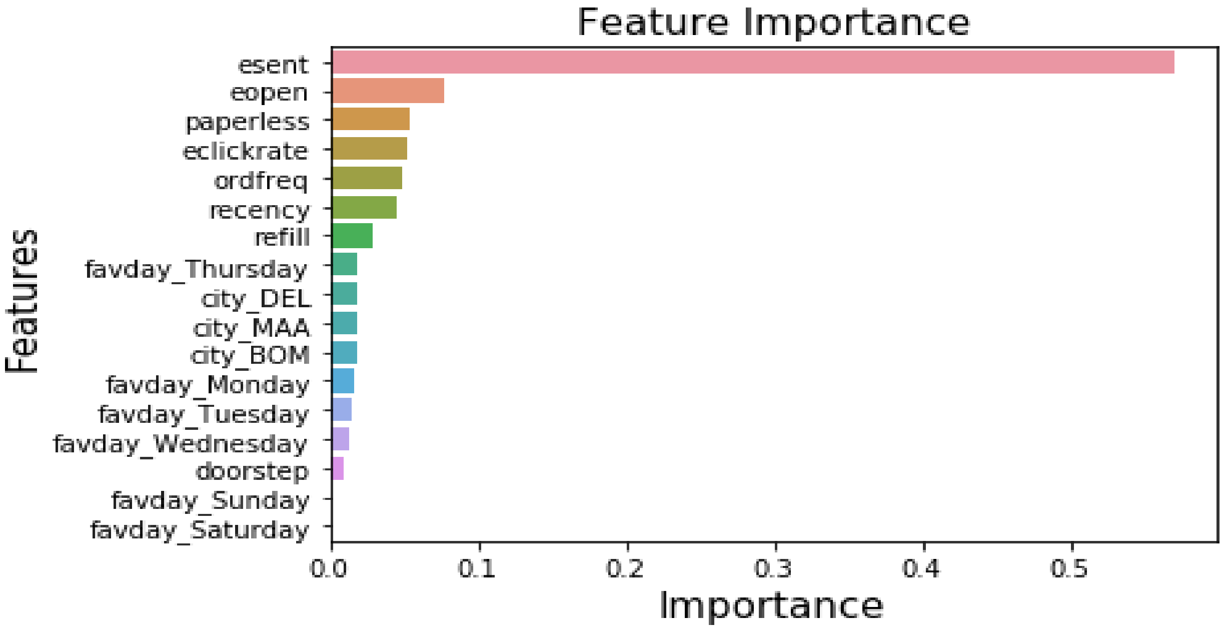
Train and Test Metrics for Minority class:

Model	Precision		Recall		F1 Score	
	Train	Test	Train	Test	Train	Test
Extreme Gradient boost	0.96	0.94	0.90	0.88	0.93	0.91
Gradient boost	0.95	0.93	0.87	0.88	0.91	0.91
Random Forest	1.00	0.94	1.00	0.86	1.00	0.90

Based on the F1 score for the majority and minority class, all three algorithms have better F1 scores such as 0.98 and 0.91 for majority and minority class respectively.

Based on other metrics such as Accuracy, Precision, Recall and ROC-AUC score, we can conclude that Extreme Gradient Boost algorithm is performing better than other algorithms.

5.1.6 Feature Importance:



‘esent’ plays a vital role in predicting the target variable ‘Retention’ and Number of mails sent is the main factor of affecting the customer retention.

CHAPTER 6

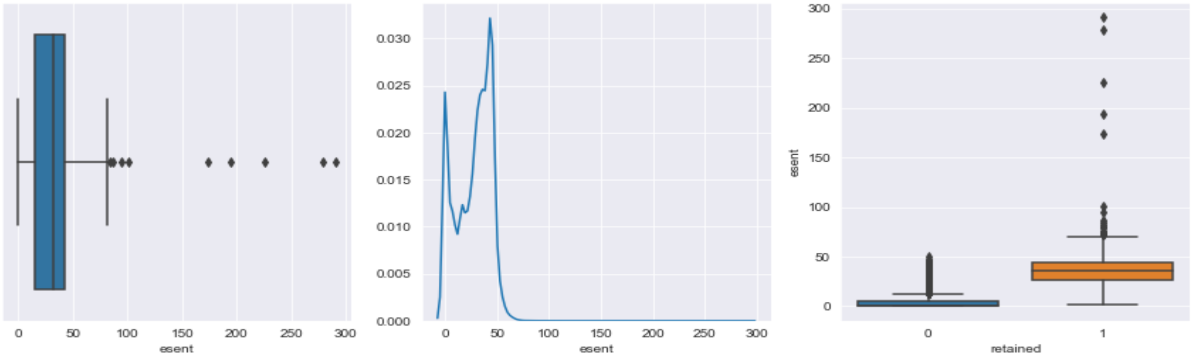
6.1 CONCLUSION:

We have applied Logistic Regression, Naïve Bayes, Decision Tree, Random Forest, Bagging Classifier, Ada Boost Classifier, Gradient Boosting Classifier and Extreme Gradient Boosting Classifier to the dataset and also tuned their hyper parameters to identify the best model that can classify the customers who might subscribe to the term deposit. The best model achieved was Extreme Gradient Boost Classifier algorithm with an accuracy of 0.9638. As per the objective this project, we can see that the feature 'esent' has more impact on target variable.

6.1.1 Closing Reflections:

As esent was most important feature for our prediction further bivariate analysis was done on esent variable and we have found the insights below.

ESENT:



- ➢ When no promotional mails were sent, none of the customers were retained.
- ➢ When more than 40 (in a range of 40 to 50) promotional mails were sent, customers were more likely to be retained.
- ➢ When more than 50 promotional mails were sent, all the customers were retained.

6.1.2 Suggestions:

In order to retain the customers, the retailers adapted various marketing strategies and market basket analysis and various other methods to study their customer shopping pattern. As per our literature survey, we have seen that the customer churn was predicted mainly based on CRM records, feedbacks from social media, feedback survey conducted on behalf of the company. But our perspective is that most of the marketing is done through primary mode of communication which is email. So, in this project, we are going to analyze whether the promotional emails sent to the customer has any effect on customer churn.

Based on the analysis, we suggest that, offers and freebies can be offered for customer who will be churned. More customised offers can be given to them. Further if we get the data on products that they have purchased, we can perform market basket analysis and provide them better recommendations by Identifying the patterns of shopping in the customers.

Literature Survey:

- https://www.kaggle.com/uttamp/store-data
- https://www.fortunebusinessinsights.com/industry-reports/organic-tea-market-100804
- https://www.expertmarketresearch.com/reports/indian-tea-market
- https://www.slideshare.net/hemanthcrpatna/a-project-report-on-retail-industry-in-india
- https://www.crazyegg.com/blog/customer-retention/
- https://www.vendhq.com/blog/customer-retention-strategies-retail/#:~:text=What%20is%20customer%20retention%3F,to%20retain%20those%20existing%20shoppers
- https://www.superoffice.com/blog/reduce-customer-churn/
- https://www.qualtrics.com/au/experience-management/customer/customer-churn/
- https://www.kdnuggets.com/2019/05/churn-prediction-machine-learning.html
- https://buildfire.com/customer-retention-strategies/
- https://neilpatel.com/blog/improve-churn-rate/
- https://www.returnlogic.com/blog/what-is-customer-churn-in-ecommerce
- https://www.salesforce.com/resources/articles/how-calculate-customer-churn-and-revenue-churn/
- https://www.kaggle.com/hellbuoy/online-retail-customer-clustering
- https://www.kaggle.com/uttamp/store-data
- https://towardsdatascience.com/hands-on-predict-customer-churn-5c2a42806266
- https://www.vendhq.com/blog/how-retailers-can-use-data-to-boost-productivity-customer-service-sales/
- https://www.ijrte.org/wp-content/uploads/papers/v8i6/F9550038620.pdf
- https://www.gainsight.com/guides/the-essential-guide-to-churn/
- Analysis of Customer Churn prediction in Logistic Industry using Machine Learning- International Journal of Scientific and Research Publications, Volume 7, Issue 11, November 2017 ISSN 2250-3153