# Maximize the customer retention by predicting the customer churn in retail


Team Members: (Group 8)

1. Vandhana Priya V
2. Sreeshitha Sreedhar
3. Tamilarasan G
4. Vijay Paudel B
5. Swaruba P

Mentored by:
Jayveer Nanda

# Problem Statement

**Domain** - Retail Industry

**Business problem statement** – Customer Churn in Retail Industry

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.

**Why to Predict Customer Churn:**

- 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%.

# DATASET

- The dataset is collected from an online tea retail store which sells tea of different flavors.
- Across 4 major cities - Bangalore, Chennai, Delhi and Mumbai.
- Dataset contains data about the store's customers, their orders, quantity ordered, order frequency, city, details of promotional mails sent to the customers etc.
- We have collected the data between 2008 and 2018.
- Rows : 30801 , Columns : 15
- The reason we chose this dataset is that, it included the details about promotional mails.

# DATA DICTIONARY

Numerical Columns : 8 , Categorical columns : 7

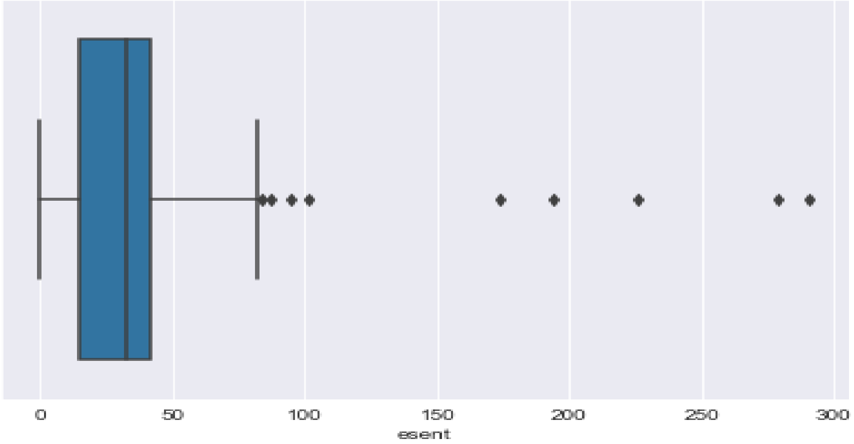
| Features | Description |
|---|---|
| **Custid** | Computer generated ID to identify customers throughout the database |
| **Retained** | 1, if customer is assumed to be active, 0 = otherwise |
| **Created** | Date when the contact was created in the database - when the customer joined |
| **Firstorder** | 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 |

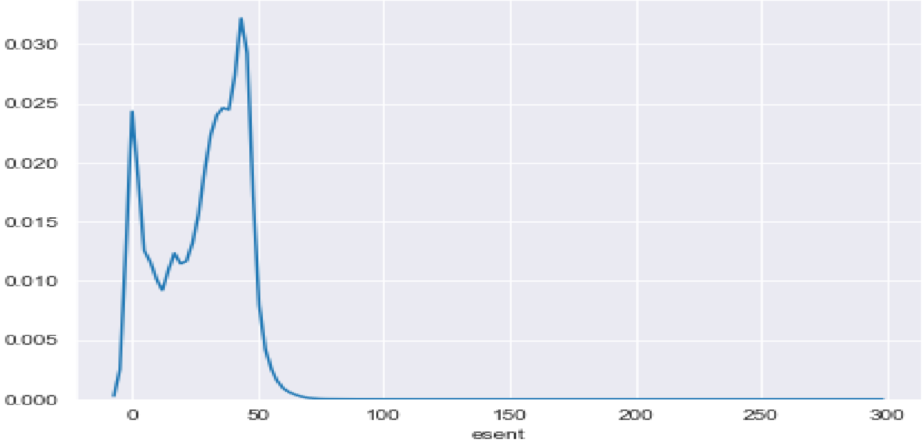
# DATA CLEANING AND PREPROCESSING

- Null value imputation
- Handling Non-Standard values and duplicated values
- Alternate sources of data that can supplement the core dataset
    - Tenure
    - Recency
    - Eopen
- As per our domain, we are having extreme values and not outliers

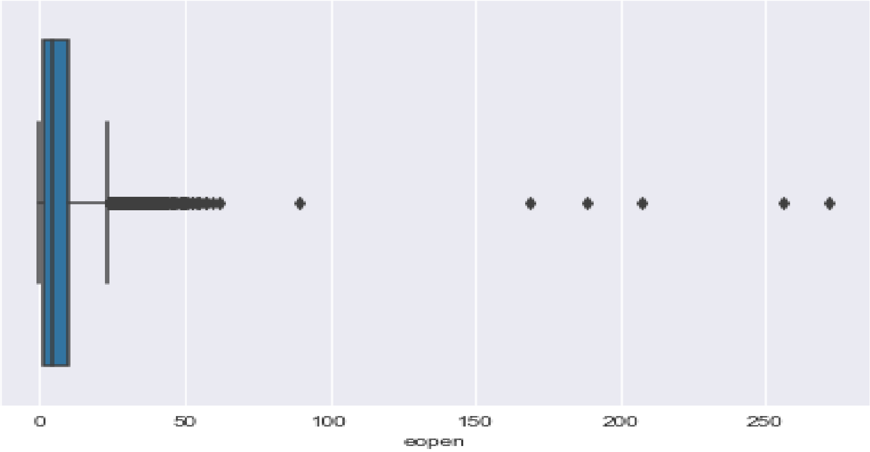
# EDA

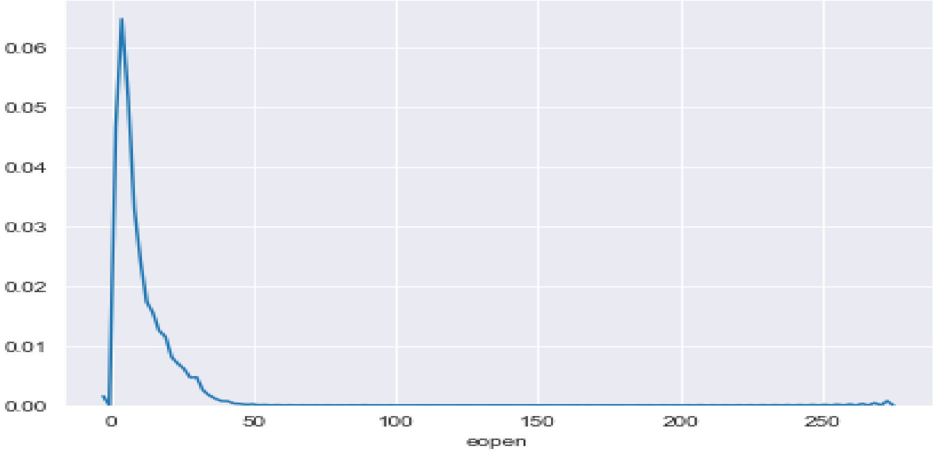
## Esent



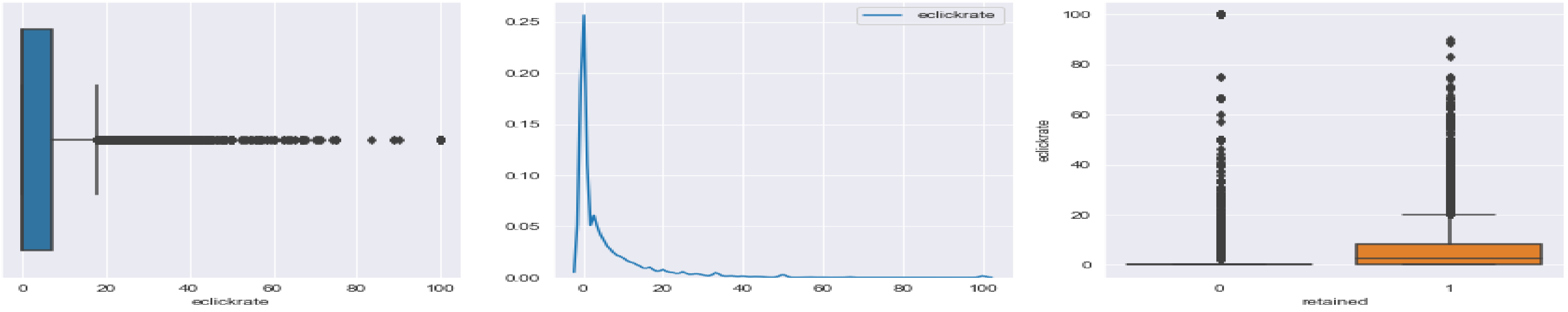




## Eopen

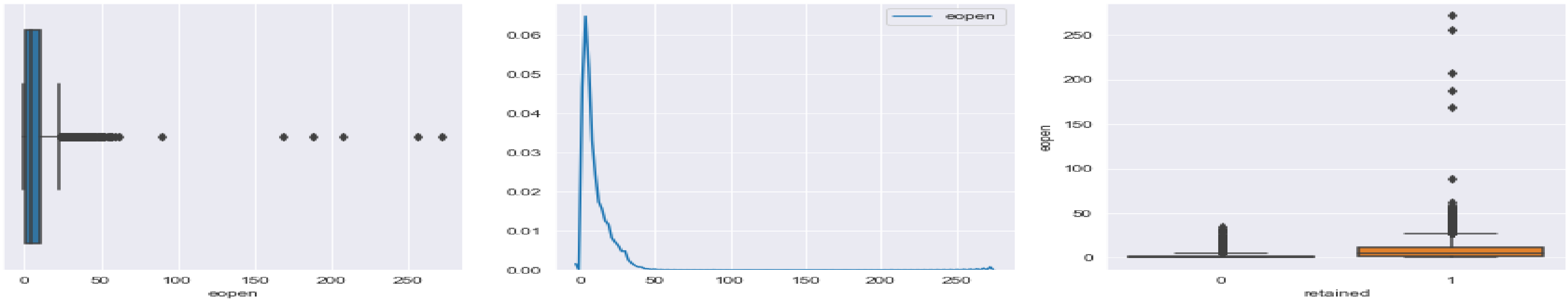


## Eclickrate Vs Retained




## Eopen vs Retained




- Studying the effect of promotion mails on customer retention

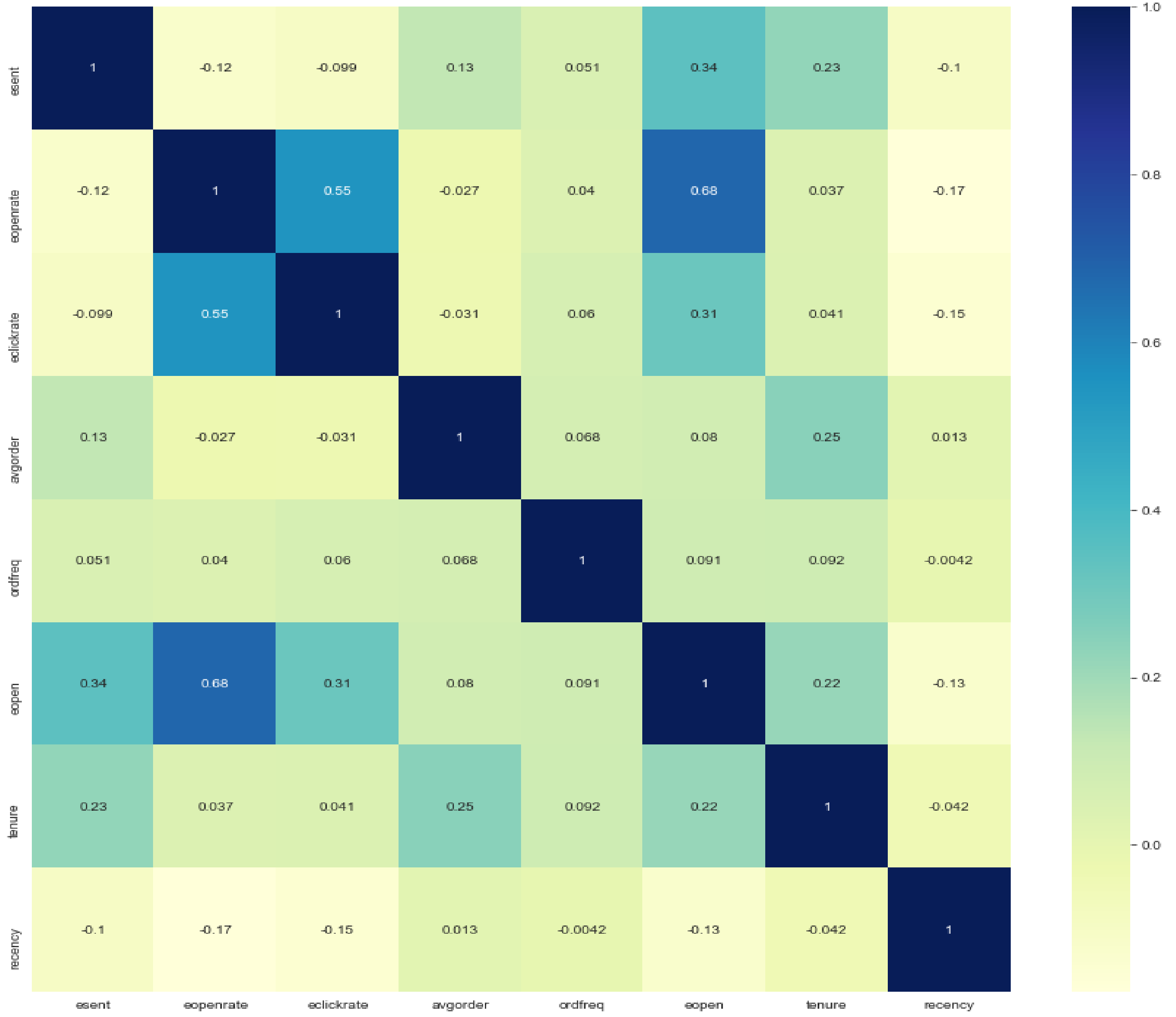
# Correlation matrix

**Interpretations:**

- Esent, eopenrate, eclickrate and tenure are collinear

**Variation Inflation Factor (VIF)** – to check the variance of the features eopenrate, eopen, esent has nearly equal to 4.5

| | 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 |



greatlearning

# Statistical significance of variables

- Tests Conducted – Shapiro Test, Mannwhitneyu Test, Proportion Ztest, Chi2 ContingencyTest
- None of the numerical features were following Normal Distribution
- Significance variables obtained from Mannwhitneyu test were esent, eopen, eopenrate, eclickrate, ordfreq, recency
- All the categorical features were significant variables.
- All numerical features are significant variables except avgorder and tenure

# Class imbalance and treatment

- Class ’1’  =  24,425 (79.46%)
- Class ‘0’ = 6,310 (20.5%)
- There is an imbalance in the dataset, but this imbalance is 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’.
- This can be treated using SMOTE technique

# Feature Engineering

- **Scaling and Transformation :**
  - Scaling -To standardize the numerical features in the data, We have used Standard scaler.
  - One hot Encoding - For the categorical features in the data
- **Feature Selection :**
  - Based on VIF and statistical tests
  - 3 Features were dropped – eopenrate, avgorder, tenure

# Model evaluation metrics

- F1 Score
- Precision
- Recall

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.

# Base Model – Evaluation Metrics

| | 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 |

- Based on the F1 score;  Extreme Gradient Boost classifier, Random Forest, Gradient Boosting and Bagging classifier gives better results comparatively.

greatlearning

# Using SMOTE Technique

| 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 |

- SMOTE model did not show much differences compared to the other models.
- There was a slight drop in the F1 scores compared to the base model.

# VIF and Statistical Tests

| 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 |

- The F1 score obtained after feature selection was comparatively similar to that of the base models.

greatlearning

# Hyper Parameter tuning

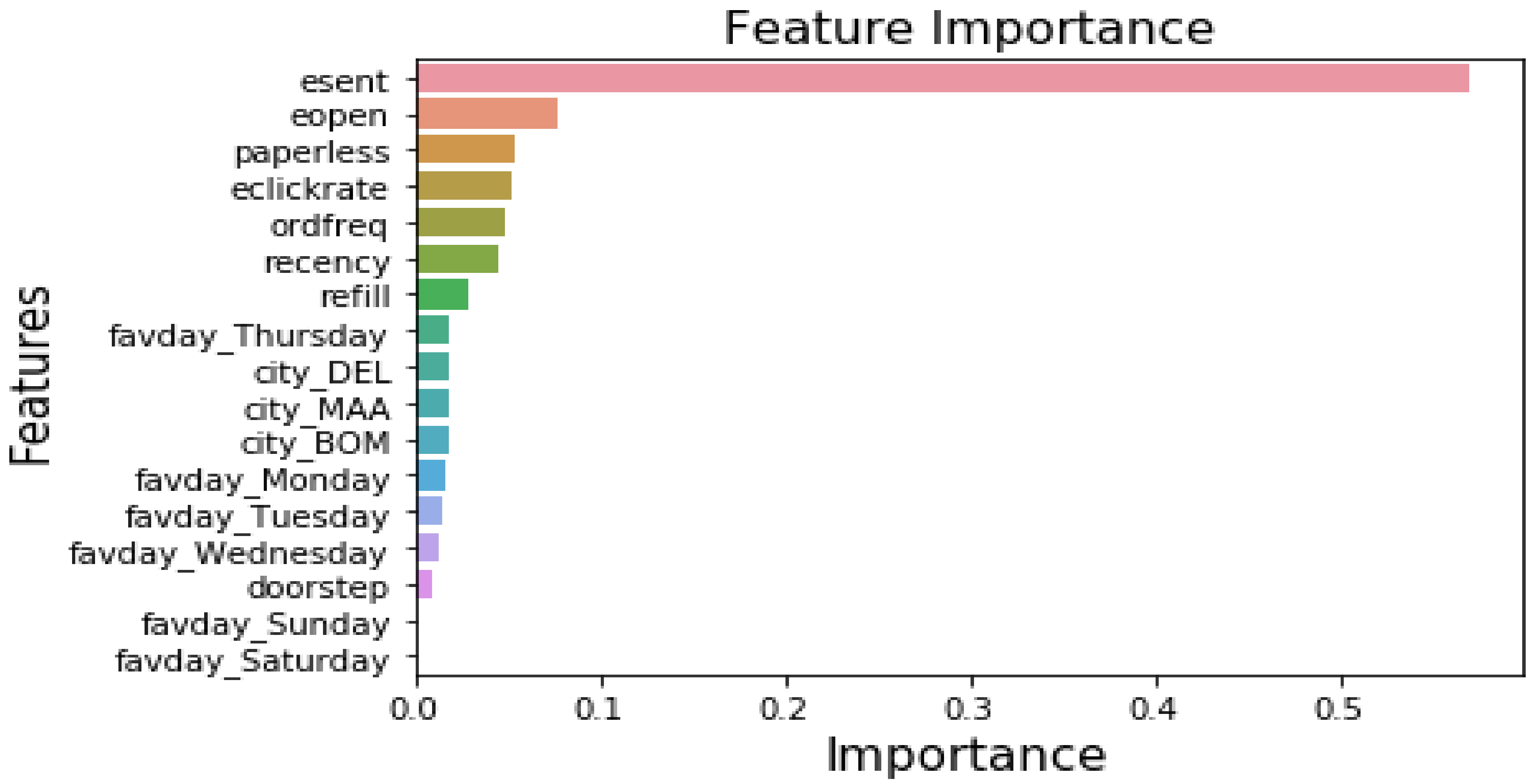
| SL.No | Model | Precision Score | Recall Score | Accuracy Score | f1-score | AUC Score |
|---|---|---|---|---|---|---|
| 0 | Extreme Gradient Boost Classifier | 0.968811 | 0.986506 | 0.963863 | 0.977578 | 0.930305 |
| 1 | Gradient boost | 0.9706 | 0.9838 | 0.9633 | 0.9772 | 0.9812 |
| 2 | Random Forest | 0.968310 | 0.985519 | 0.962681 | 0.976839 | 0.928833 |
| 3 | Decision Tree | 0.967846 | 0.985684 | 0.962418 | 0.976684 | 0.927936 |
| 4 | Bagged Decision Tree | 0.958778 | 0.991279 | 0.959001 | 0.974757 | 0.911164 |
| 5 | Ada boost Decision Tree | 0.938283 | 0.998190 | 0.946124 | 0.967310 | 0.868958 |
| 6 | Logistic Regression | 0.9563 | 0.9665 | 0.9381 | 0.9614 | 0.9658 |
| 7 | Naïve Bayes Algorithm | 0.9408 | 0.934 | 0.9408 | 0.9374 | 0.8506 |
| 8 | Bagged Naïve Bayes Algorithm | 0.9426 | 0.9319 | 0.9003 | 0.9372 | 0.8534 |

# Best model :  Extreme Gradient Boost

| Sl.No | Model | Precision Score | Recall Score | Accuracy Score | f1-score | AUC Score |
|---|---|---|---|---|---|---|
| **0** | Extreme Gradient Boost Classifier | 0.968811 | 0.986506 | 0.963863 | 0.977578 | 0.930305 |
| **1** | Gradient Boost | 0.9706 | 0.9838 | 0.9633 | 0.9772 | 0.9812 |
| **2** | Random Forest | 0.968310 | 0.985519 | 0.962681 | 0.976839 | 0.928833 |

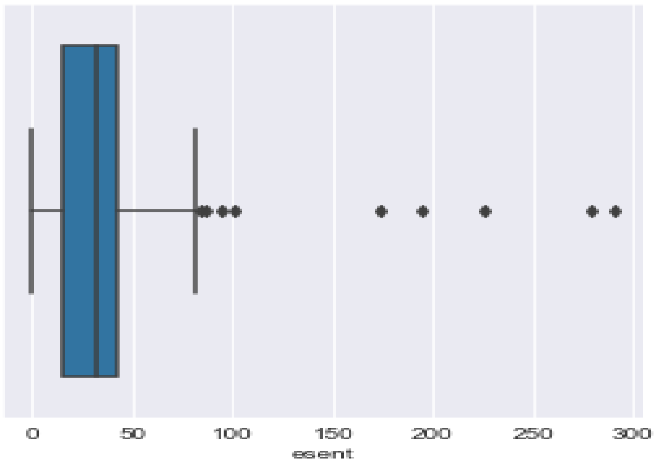
➢Based on metrics such as the majority and minority class F1 score, Accuracy, Precision, Recall and ROC-AUC score, We can conclude that Extreme Gradient Boost algorithm is performing better than other algorithms.

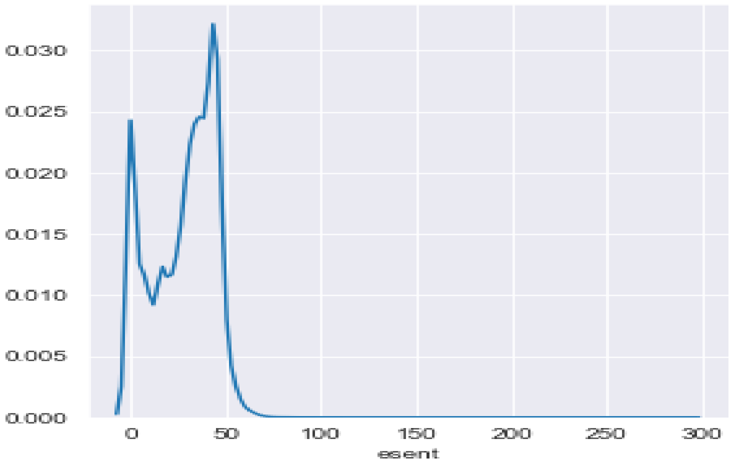
# Conclusion

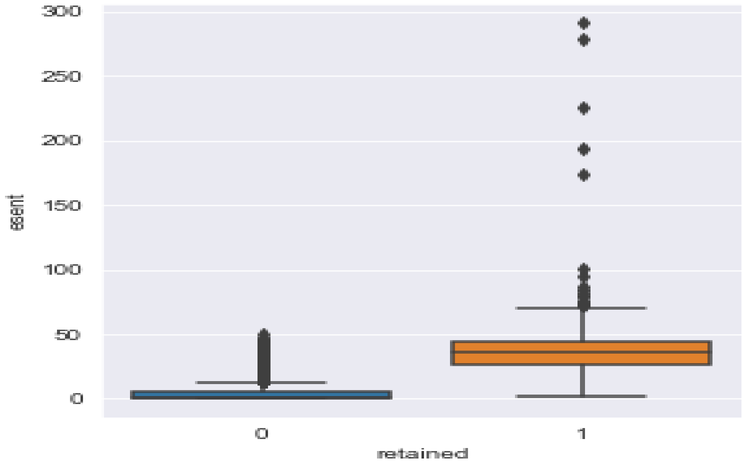


Feature Importance
esent
eopen
paperless
eclickrate
ordfreq
recency
refill
favday_Thursday
city_DEL
city_MAA
city_BOM
favday_Monday
favday_Tuesday
favday_Wednesday
doorstep
favday_Sunday
favday_Saturday
Features
0.0
0.1
0.2
0.3
0.4
0.5
Importance

# Esent




- 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

# THANK YOU