Project Summary

Batch details	PGP DSE(Online) – FEB'21
Team members	Ashwini R Jannu, Sreerag, Anetha Ranganathan, Akash Sah, Geetha k
Domain of Project	Marketing and Retail Analytics
Proposed project title	Customer Churn Prediction
Group Number	1
Team Leader	Ashwini R Jannu
Mentor Name	Mr Jayveer Nanda

Date: 11th Sept, 2021

Signature of the Mentor

Jayveer Nanda

Signature of the TL

Ashwini R

Table of Contents

Project Details

INDUSTRY REVIEW

The purpose of this study is to predict the customer churn within a retail industry. Customer churn is an important metrics for a growing business to evaluate. Customer churn occurs when customers / subscribers stop doing business with a company or a service. Also known as customer attrition, customer churn is a critical metric because it is much less expensive to retain existing customers than to acquire new customers – earning business from new customers means working leads all the way through the sales funnel, utilizing your marketing and sales resources throughout the process. Customer retention, on the other hand, is generally cost-effective as you've already earned the trust and loyalty of existing customers.

BUSINESS PROBLEM STATEMENT (GOALS)

1. Business Problem Understanding

   A top priority in any business is a need to increase the revenue and the profits. One of the causes for a decrease in profits is when current customers stop transacting. When a customer leaves / churns from a business, the opportunity for potential or future sales or cross selling is lost. If a customer leaves the business without any form of advice, the company may find it hard to respond and take corrective measures. Ideally companies should adopt a proactive and identify potential churners prior to them leaving.

2. Business Objective

   Customer retention strategies have been a less costly alternative than attracting new customers. Through data available within the Point of Sales systems, customer transactions may be extracted and their buying patterns may be analyzed. Customer churn is has a significant impact on a business, however it is noted that 2 out of 3 companies have no strategy for preventing customer churn.

   Churn is a major problem for many companies as it shows how good or bad they are at keeping customers by their side. A survey says that it costs **5 times more** to acquire new customers than it does to keep an existing one. It will cost **16 times more to bring a**

new customer up to the same level as an existing customer. The second reason lies in the fact that the more customers a business retains, the more revenue it makes.
For example, the Harvard Business School report says that on an average, **5% increase in customer retention** rates results in **25% -95% increase of profits.** The same is revealed by KPMG, who found that customer retention is the main driver of a company's revenue**.**
The below image shows the significant retail revenue drivers.

Through our project we wish to predict churn within a retail industry. The data used here is the data of an online tea retail store which sells tea of different flavors across various cities in India.

1) Literature Survey :

There are significant amount of works carried out in customer churn in various domains using different classification algorithms. Some of their works are as follows:

- Abinash Mishra and U. Srinivasulu Reddy performed a comparative study on ensemble methods. The performance metrics calculated here are accuracy, error rate, specificity (portion of negative cases that were classified correctly) and sensitivity (portion of positive cases that correctly identified). The result show that Random Forest algorithm performed better with 91.66% low error rate of less specificity 53.54 and high sensitivity 98.89 than others like Bagging, Boosting and Decision tree.
- Pretam Jayaswal, Bakshi Rohit Prasad, Divya Tomar, and Sonali Agarwal selected Bagging and Boosting based ensembles classifiers like Random Forest, Gradient Boost Trees (GBT) and Decision tree classifier where dataset is divided in training (75%) and testing (25%) subsets using Apache Spark. This work shows that GBT outperformed other methods in terms of accuracy of 96.78% and specificity. They also employed optimization phase that made the results more accurate and refined.
- Essam Shaaban, Yehia Helmy, Ayman Khedr, Mona Nasr proposed a model that predicts churners into 3 categories in using a retention strategy. They have used decision tree, Support vector machine and Neural Network through an open source software called WEKA. The data set is divided into training set of 80% instances and the testing set of 20% instances. The training data contains 80% instances are categorized as non-churners and others are 20% are categorized as churners. In this paper, both SVM and Neural networks showed the same results in terms of accuracy

and error rates.

- Irfan Ullah, Basit Raza, Ahmad Kamran Malik, Muhammad Imran, Sail Ul Islam, Sung Won Kim proposed a hybrid model using Random Forest, Decision Stump, J48 and Random Tree with 10-fold cross-validation for classification of churners and non-churners and used k-means for clustering. The proposed model targets churn customers and distinguish the explanations for their relocation. The experimental result 88.63% correct classification through RF. In this study, for factor identification, a comparable classifier such as Attribute Selected Classifier is used for rule generation that can be easily visualized since RF is not appropriate for rule generation for factor identification as it generates complex forest which is difficult to visualize and rule inference.
- Jaya Kawale, Aditya Pal, Jaideep Srivastava proposed an updated diffusion model in online role-playing games (MMORPG) where they considered two aspects-decrease in player engagement of churners over time until they churn and increase in churn prosperity with increase in number of churner neighbors. Using diffusion model having two valued influence vectors in modelling accurately- negative influence and positive influence and a parameter spread factor where the portion of influence transferred to his network. Their technique outperformed both diffusion model and network and engagement feature based classification in their dataset. This scheme is able to capture social influence and player engagement effectively and resulted in a consistent prediction accuracy. However, they did not compare their approach with classification based on features that might not capture both important aspects mentioned.

2) Dataset and Domain

Original owner of data	Uttam P
Data set information	The dataset is of an online tea retail store which sells tea of different flavors across various cities in India. It contains data about the store's customers, their orders, quantity ordered, order frequency, city etc.
Number of rows and columns	30801 & 15
Domain	Retail and Marketing
Link to web page	https://www.kaggle.com/uttamp/store-data

- A dataset of an online tea retail store which sells tea of different flavors across various cities in India is used here in predicting customer churn.
- This dataset can be used to understand what are the various marketing strategy based on consumer behaviour that can be adopted to increase customer retention of a retail store.
- The dataset has 30801 rows and 15 columns
- These are the columns present in the dataset :

1. custid : Referring to the customer id of the customers
2. retained : 1, if customer is assumed to be active, 0 = otherwise
3. created : Date when the contact was created in the database - when the customer joined
4. firstorder : Date when customer made the first order
5. lastorder : Date when customer made the last order
6. esent : Number of emails sent
7. eopenrate : Number of emails opened divided by number of emails sent
8. eclickrate : Number of emails clicked divided by number of emails sent
9. avgorder : Average order size for the customer
10. orderfreq : frequency of orders
11. paperless : 1 if customer subscribed for paperless communication (only online)
12. refill : 1 if customer subscribed for automatic refill
13. doorstep : 1 if customer subscribed for doorstep delivery
14. favday : Customer's favorite delivery day
15. city : city the customer belongs to

3. Data Exploration(EDA)

The main focus of our project is to predict the customer churn in a retail industry. We do the following EDA and Pre-processing techniques to get the data ready for model building and predicting.

DATA ANALYSIS, CLEANING/ PREPROCESSING: The pre-processing of the dataset before performing ML functions involves the following:

1) Data Wrangling : Understanding Data Types, checking for missing values, checking if the data is in the correct format.
    - Data types : We could see that the datatype for column 4 and 5("firstorder" and "lastorder") was incorrect, hence we had to change it from ‘object’ to ‘datetime’

```
1  df.dtypes
```

```
custid                object
retained               int64
created       datetime64[ns]
firstorder            object
lastorder             object
esent                  int64
eopenrate            float64
eclickrate           float64
avgorder             float64
ordfreq              float64
paperless              int64
refill                 int64
doorstep               int64
favday                object
city                  object
dtype: object
```

- Missing values/ Null values : We observed that 4 columns namely ‘custid’, ‘created’, ‘firstorder’, ‘lastorder’ had 20 missing values which we have been dropped.

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

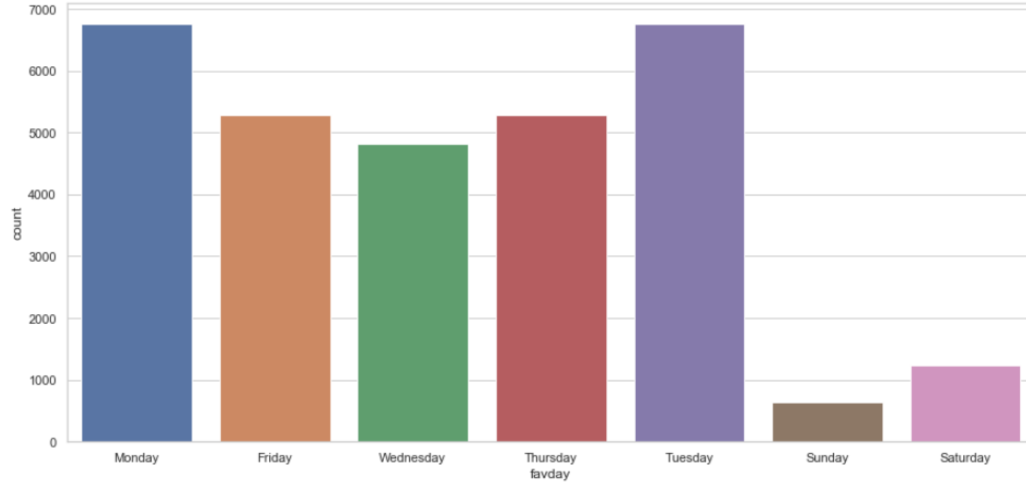
```
custid        20
retained       0
created       20
firstorder    20
lastorder     20
esent          0
eopenrate      0
eclickrate     0
avgorder       0
ordfreq        0
paperless      0
refill         0
doorstep       0
favday         0
city           0
dtype: int64
```

- Formatting : We could observe a few absurd entries in the columns ‘firstorder’ and ‘lastorder’ which had to be formatted.

2) Data Visualization :

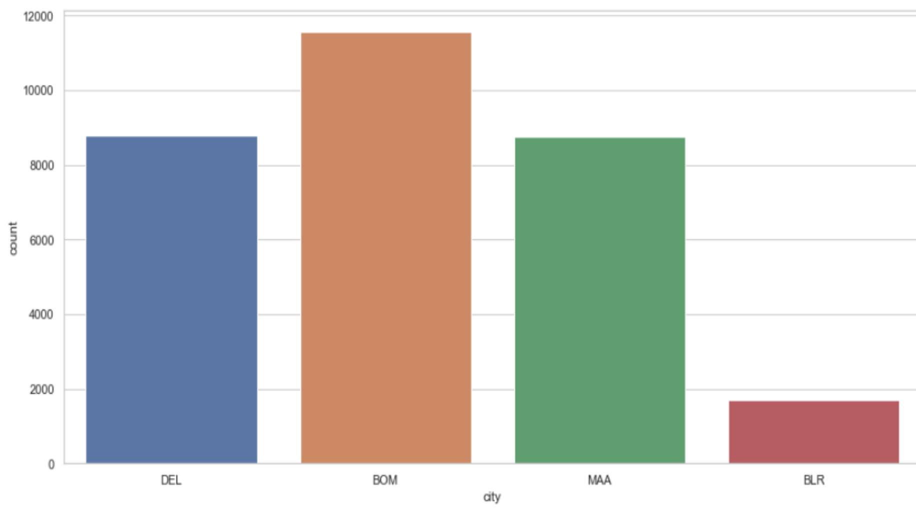
<u>Univariate Analysis :</u>

- The below shows the plot of the column ‘favday’.



Insight : The graph talks about the favourite delivery days of the customers. We can see that most of the customers choose their delivery day over the weekdays(Monday and Friday being the highest) than over the weekends.

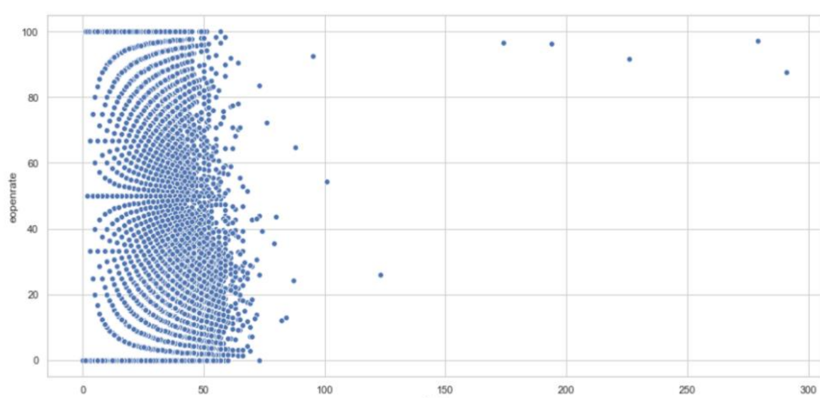
- The below shows the plot of the column ‘city’.



Insight : The graph talks about the city where the customers belong to. From the graph, we can say that there are highest number of customers from Mumbai

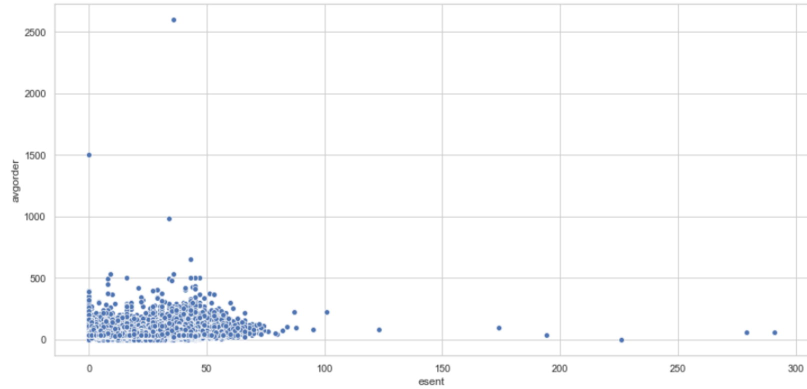
Bivariate Analysis :

- The below shows the plot of the column ‘esent’ vs ‘eopenrate’.



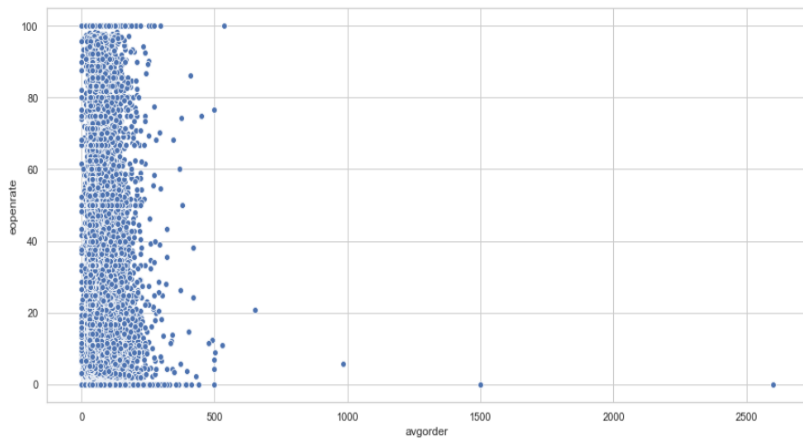
Insight: Most of the values of eopenrate lies less than esent = 80.

- The below shows the plot of the column ‘esent’ vs ‘avgorder’.



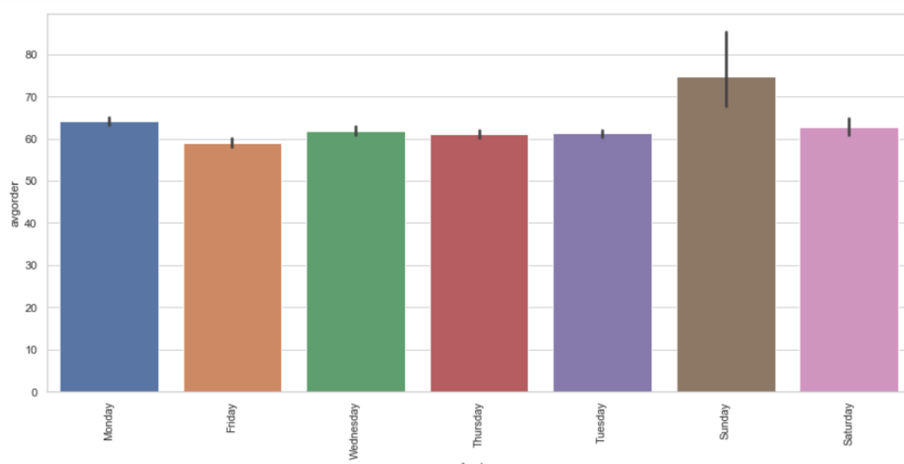
Insight: Most of the values of eopenrate lies less than esent = 80.

- The below shows the plot of the column ‘avgorder’ vs ‘eopenrate’



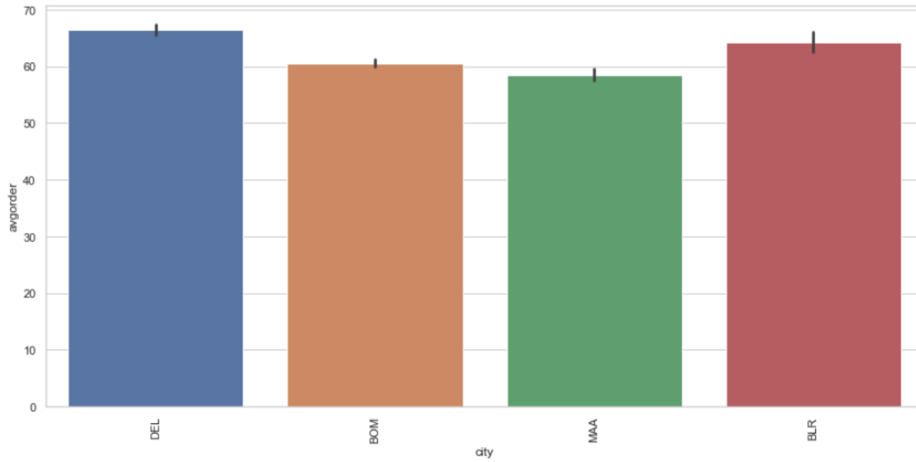
Insight: Most of the values of avgorder lies between 0 to 500

- The below shows the plot of the column ‘favday’ vs ‘avgorder’



Insight: Customers with favourite delivery day as sunday have slightly more average orders as compared to the others.

- The below shows the plot of the column 'city' vs 'avgorder'



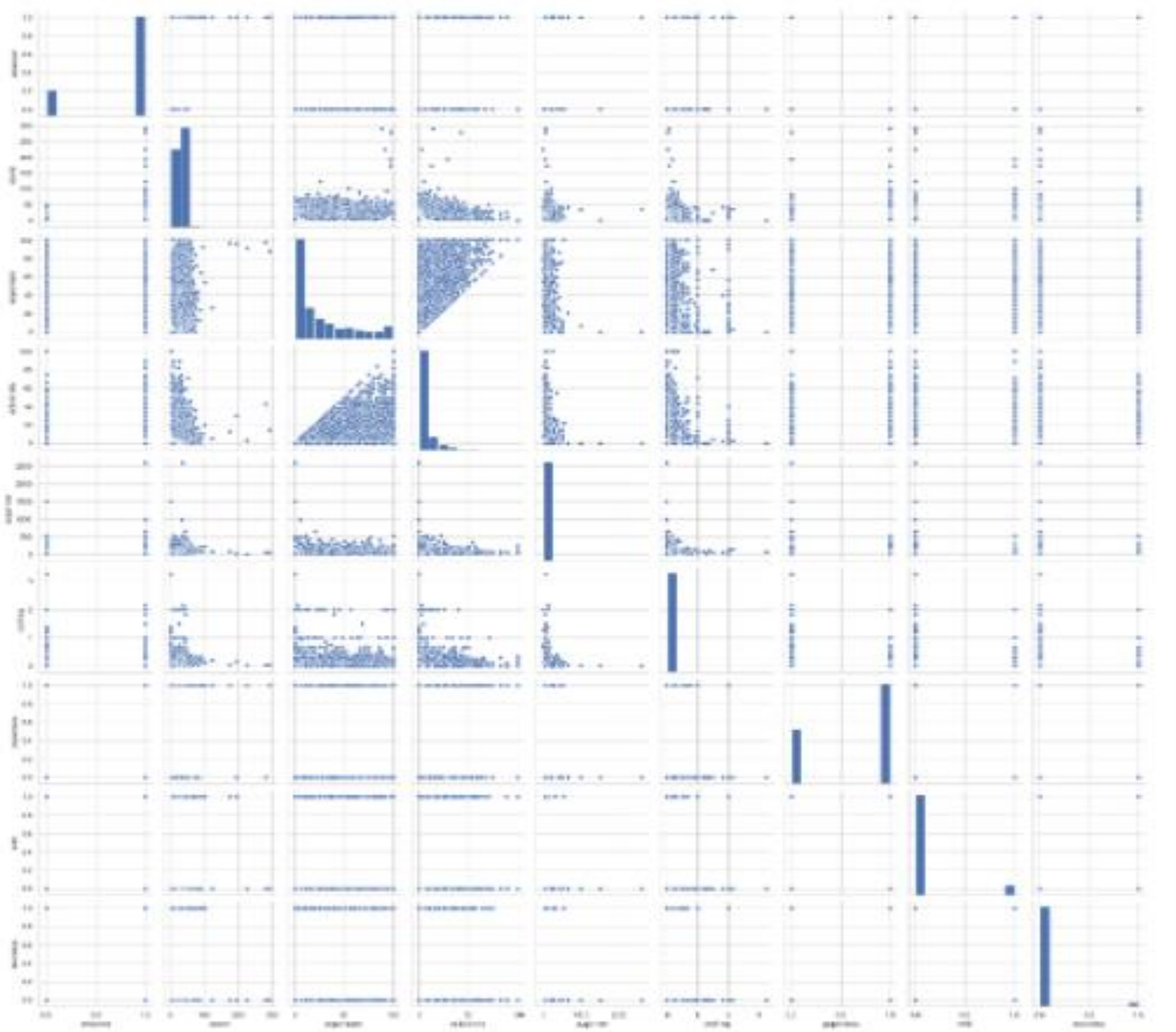
Insight : From the above plot we can infer that customers from Bangalore and Delhi have the highest avgorder as compared to other cities

<u>Multivariate Analysis :</u>

- The below plot shows the boxplot.

Insight : We could observe slight correlation between independent variables.

- The below is the pairplot which shows the relationship between the numeric variable.



3) Feature Engineering

Outliers

Outliers are extreme values that fall outside of the observations. There are many techniques to identify these outliers like outlier modelling such as IQR and Z-score method.

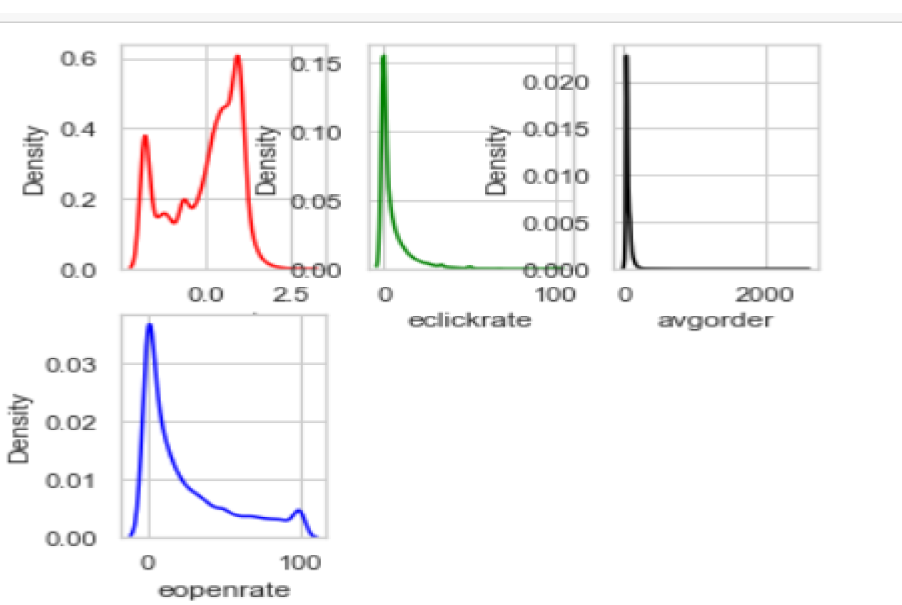
Identification and treating of the outliers are important because: -

1) Outlier may be indicted as bad data.
2) We cannot determine if the outlier data is an outlier or not.
3) Outliers can affect the final model result.
4) outlier affects both statistical results and the assumptions.

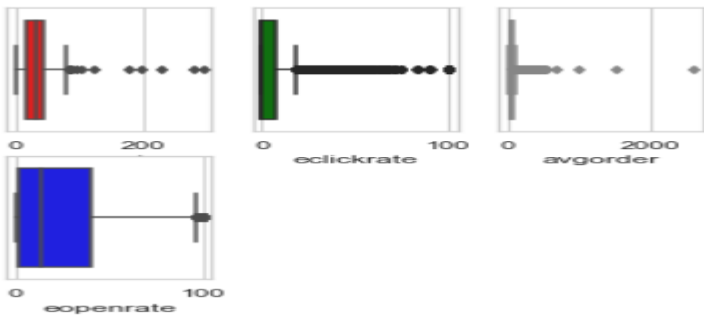
In our dataset we have outlier values for the following columns
Esent, eclickrate, avgorder, eopenrate. The following kdeplot and bar plot will show the outliers for the above mentioned columns.

Outlier Treatment

```
plt.figure(figsize=(15,10))
plt.subplot(2,3,1)
sns.kdeplot(xtrain.esent,color='red');
plt.subplot(2,3,2)
sns.kdeplot(data.eclickrate,color='green')
plt.subplot(2,3,3)
sns.kdeplot(data.avgorder,color='black')
plt.subplot(2,3,4)
sns.kdeplot(data.eopenrate,color='blue')
plt.show()
```



```
plt.figure(figsize=(15,10))
plt.subplot(2,3,1)
sns.boxplot(x = data.esent,color='red');
plt.subplot(2,3,2)
sns.boxplot(x = data.eclickrate,color='green')
plt.subplot(2,3,3)
sns.boxplot(x = data.avgorder,color='pink')
plt.subplot(2,3,4)
sns.boxplot(data.eopenrate,color='blue')
plt.show()
```



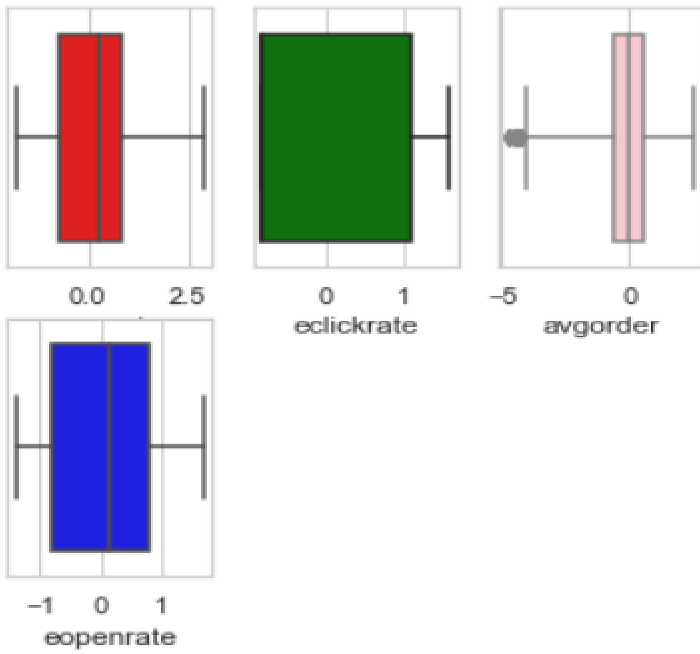
Here we are treating the outliers by using the IQR method. We chose the data between the Upper limit and Lower limit.

```
def impute(x):
    for i in x.columns:
        iqr = 1.5*(x[i].quantile(0.75) - x[i].quantile(0.25))
        ul = x[i].quantile(0.75) + iqr
        ll = x[i].quantile(0.25) - iqr
        temp = []
        for j in x[i].index:
            if x[i][j] > ul or x[i][j] < ll :
                temp.append(np.median(x[i]))
            elif x[i][j] < 0:
                temp.append(0)
            else:
                temp.append(x[i][j])
        x[i]= temp
    return x
```

```
xtrain[['esent','eclickrate','avgorder','eopenrate']] = impute(xtrain[['esent','eclickrate','avgorder',
xtest[['esent','eclickrate','avgorder','eopenrate']] = impute(xtest[['esent','eclickrate','avgorder','e
```

Activate Windows
Go to Settings to activate

```
plt.figure(figsize=(5,5))
plt.subplot(2,3,1)
sns.boxplot(x = xtrain.esent,color='red');
plt.subplot(2,3,2)
sns.boxplot(x = xtrain.eclickrate,color='green',whis=6)
plt.subplot(2,3,3)
sns.boxplot(x = xtrain.avgorder,color='pink',whis=3)
plt.subplot(2,3,4)
sns.boxplot(xtrain.eopenrate,color='blue')
plt.show()
```



Bar plot shows the columns after treating the null values. We removed the outliers and data is ready without any extreme values.

<u>Encoding :</u>

One hot encoding is a technique of converting data to prepare data for an algorithm and get a better prediction. With one-hot,we convert each categorical value into a column and assign a binary value of 1 or 0 to those columns.

One Hot Encoding

```
xtrain = pd.concat((xtrain.drop(columns=['city','favday'],axis=1),
                    pd.get_dummies(xtrain[['favday','city']],drop_first=True)),axis=1)
```

```
xtrain.head()
```

	esent	eopenrate	eclickrate	avgorder	paperless	refill	doorstep	favday_Monday	favday_Saturday	favday_Sunday	fav
26377	0	0.000000	0.000000	50.72	0	0	0	0	0	0	
12904	45	4.444444	2.222222	74.76	1	0	0	0	0	0	
19672	27	85.185185	14.814815	55.07	1	1	0	0	0	0	
18241	34	0.000000	0.000000	46.17	1	0	0	0	0	0	
2056	55	18.181818	1.818182	129.96	1	0	0	1	0	0	

```
xtrain.head()
```

Monday	favday_Saturday	favday_Sunday	favday_Thursday	favday_Tuesday	favday_Wednesday	city_BOM	city_DEL	city_MAA
0	0	0	0	1	0	0	1	0
0	0	0	0	0	0	0	0	1
0	0	0	0	0	1	1	0	0
0	0	0	0	1	0	0	0	1
1	0	0	0	0	0	0	1	0

```
xtest = pd.concat((xtest.drop(columns=['city','favday'],axis=1),
                   pd.get_dummies(xtest[['favday','city']],drop_first=True)),axis=1)
```

```
xtest.head()
```

Monday	favday_Saturday	favday_Sunday	favday_Thursday	favday_Tuesday	favday_Wednesday	city_BOM	city_DEL	city_MAA
1	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	1
0	0	0	0	0	1	1	0	0
0	0	0	1	0	0	1	0	0
0	0	0	0	1	0	0	0	0

```
data.dtypes
```

```
retained          int64
esent             int64
eopenrate       float64
eclickrate      float64
avgorder        float64
paperless         int64
refill            int64
doorstep          int64
favday           object
city             object
dtype: object
```

We did transfer the ‘city’ and ‘Favday’ columns using one hot encoding technique.

Transformation :

Feature Transformation is a technique of modifying the data but keeping the information. This modification will make ML algorithm to understand better and give better results, so that we can reduce data repetition and improve data efficiency.

Box cox Transformation

This technique is to transform the non-normal dependent variables in to a normal shape. Box-cox will only work for positive values and greater than 0.

If w is our transformed variable and y is our target variable, then

$$w_t = \begin{cases} \log(y_t) & \text{if } \lambda = 0; \\ (y_t^\lambda - 1)/\lambda & \text{otherwise.} \end{cases}$$

Photo from Rob Hyndman’s and George Athanasopoulos’s “Forecasting”.

where t is the time period and lambda is the parameter that we choose (you can perform the Box-Cox transformation on non-time series data, also).

For our dataset we did box cox transform for ‘avgorder’, ’esent’, ’eopenrate’, ’eclickrate’ columns.

```
xtrain['avgorder'] = s.boxcox(xtrain.avgorder+1)[0]
xtrain['esent'] = s.boxcox(xtrain.esent+1)[0]
xtrain['eopenrate'] = s.boxcox(xtrain.eopenrate+1)[0]
xtrain['eclickrate'] = s.boxcox(xtrain.eclickrate+1)[0]
```

```
xtrain.head()
```

	esent	eopenrate	eclickrate	avgorder	paperless	refill	doorstep	favday	city
26377	0.000000	0.000000	0.000000	10.050001	0	0	0	Tuesday	DEL
12904	28.737037	1.927733	0.788593	12.201272	1	0	0	Friday	MAA
19672	18.503292	6.326022	1.192450	10.476954	1	1	0	Wednesday	BOM
18241	22.578927	0.000000	0.000000	9.580403	1	0	0	Tuesday	MAA
2056	34.135013	3.710895	0.728382	10.072085	1	0	0	Monday	DEL

```
xtest['avgorder'] = s.boxcox(xtest.avgorder+1)[0]
xtest['esent'] = s.boxcox(xtest.esent+1)[0]
xtest['eopenrate'] = s.boxcox(xtest.eopenrate+1)[0]
xtest['eclickrate'] = s.boxcox(xtest.eclickrate+1)[0]
```

```
xtest.head()
```

	esent	eopenrate	eclickrate	avgorder	paperless	refill	doorstep	favday	city
23814	7.332319	0.000000	0.000000	8.233302	1	0	0	Monday	DEL
16885	27.665769	2.479603	0.982111	12.569997	1	0	0	Monday	MAA
28714	25.829369	0.000000	0.000000	9.450932	1	0	0	Wednesday	BOM
3392	29.484299	2.395254	0.000000	3.630310	0	0	0	Thursday	BOM
1764	31.286335	1.249433	0.770719	14.207098	0	0	0	Tuesday	BLR

```
xtest.describe()
```

	esent	eopenrate	eclickrate	avgorder	paperless	refill	doorstep
count	9235.000000	9235.000000	9235.000000	9235.000000	9235.000000	9235.000000	9235.000000
mean	19.850333	2.818226	0.396382	10.911535	0.648511	0.095939	0.042122
std	11.009007	2.041551	0.489902	2.351381	0.477461	0.294524	0.200879
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	12.286070	0.000000	0.000000	9.450932	0.000000	0.000000	0.000000
50%	23.350213	3.115985	0.000000	10.749145	1.000000	0.000000	0.000000
75%	28.880018	4.526993	0.951031	12.249096	1.000000	0.000000	0.000000
max	50.835025	6.319732	1.177551	16.968334	1.000000	1.000000	1.000000

```
xtrain.describe()
```

	esent	eopenrate	eclickrate	avgorder	paperless	refill	doorstep
count	21546.000000	21546.000000	21546.000000	21546.000000	21546.000000	21546.000000	21546.000000
mean	18.475017	2.974179	0.411344	10.214720	0.649355	0.094681	0.037548
std	10.320500	2.132582	0.503266	2.143288	0.477183	0.292781	0.190104
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	11.097454	1.242879	0.000000	8.901619	0.000000	0.000000	0.000000
50%	21.428730	3.268096	0.000000	10.072085	1.000000	0.000000	0.000000
75%	27.083059	4.735864	0.975280	11.487842	1.000000	0.000000	0.000000
max	48.030764	6.604207	1.214238	15.717931	1.000000	1.000000	1.000000

Scaling :
Feature scaling is important because this will normalize the range of all features so each feature contributes approximately proportionally equal. Scaling helps to standardize the data to a fixed range. If scaling is not done the ML algorithm tends to weigh greater values regardless the unit of the values.
Here we did standardize the ‘eclickrate’, eopenrate, avgorder, esent columns.

Scaling

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

```
a = SS.fit_transform(xtrain[['eclickrate','eopenrate','eclickrate','avgorder','esent']])
```

```
xtrain['eclickrate'] = a[:,0]
xtrain['eopenrate'] = a[:,1]
xtrain['eclickrate'] = a[:,2]
xtrain['avgorder'] = a[:,3]
xtrain['esent'] = a[:,4]
```

```
xtrain.head()
```

	esent	eopenrate	eclickrate	avgorder	paperless	refill	doorstep	favday	city
26377	-1.790170	-1.394670	-0.817369	-0.076855	0	0	0	Tuesday	DEL
12904	0.994357	-0.490706	0.749618	0.926893	1	0	0	Friday	MAA
19672	0.002740	1.571767	1.552109	0.122354	1	1	0	Wednesday	BOM
18241	0.397656	-1.394670	-0.817369	-0.295962	1	0	0	Tuesday	MAA
2056	1.517403	0.345465	0.629975	-0.066551	1	0	0	Monday	DEL

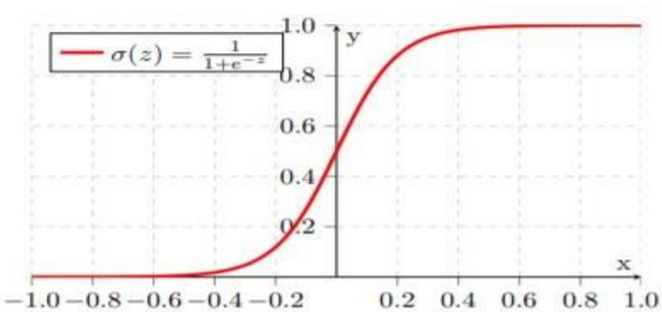
```
xtest['eclickrate'] = b[:,0]
xtest['eopenrate'] = b[:,1]
xtest['eclickrate'] = b[:,2]
xtest['avgorder'] = b[:,3]
xtest['esent'] = b[:,4]
```

```
xtest.head()
```

	esent	eopenrate	eclickrate	avgorder	paperless	refill	doorstep	favday	city
23814	-1.079691	-1.394670	-0.817369	-0.924498	1	0	0	Monday	DEL
16885	0.890554	-0.231919	1.134151	1.098934	1	0	0	Monday	MAA
28714	0.712613	-1.394670	-0.817369	-0.356371	1	0	0	Wednesday	BOM
3392	1.066764	-0.271473	-0.817369	-3.072179	0	0	0	Thursday	BOM
1764	1.241375	-0.808778	0.714102	1.862779	0	0	0	Tuesday	BLR

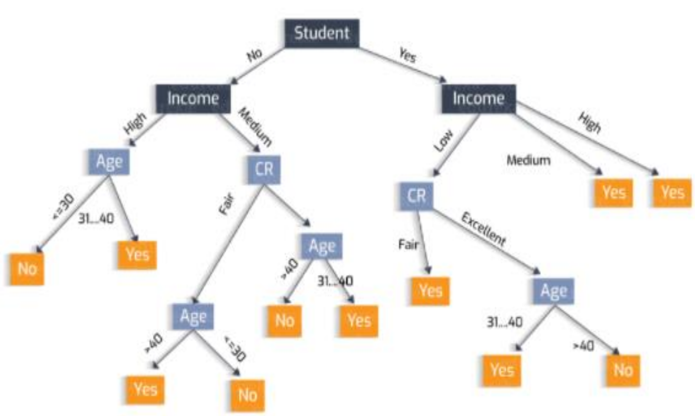
5) Assumptions :

Logistic Regression



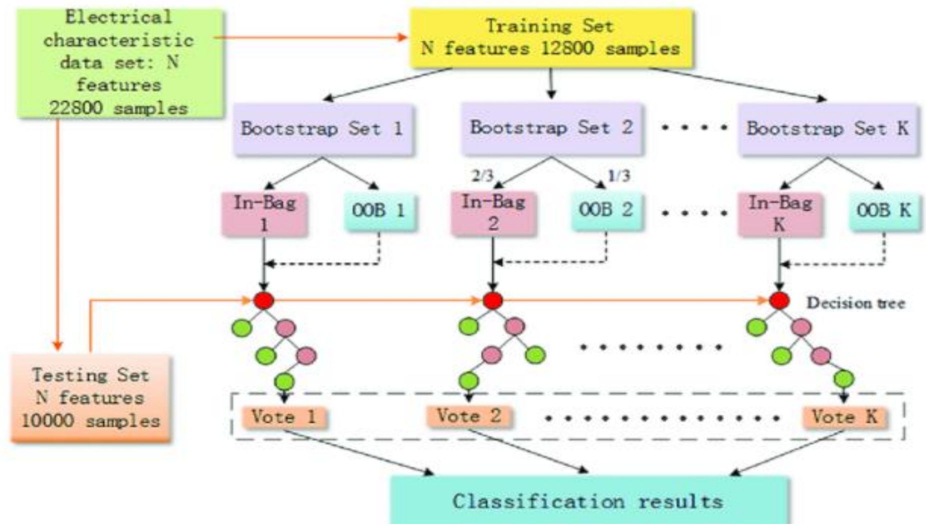
- It assumes that there is minimal or no multicollinearity among the independent variables.
- It usually requires a large sample size to predict properly.
- It assumes the observations to be independent of each other.

Decision Trees :



- Initially, whole training data is considered as root.
- Records are distributed recursively on the basis of the attribute value.

Random Forest :



- Assumption of no formal distributions. Being a non-parametric model, it can handle skewed and multi-modal data.

Next Steps :

- As the next steps, we would be trying out other models like Decision Tree, Random Forest, Gradient Boost and Neural Networks.
- We would be comparing the models and the best model will be selected.

REFERENCES

The references can be blogs, articles or even social media news relevant to explain the importance of the projects.

- Gradient Boosting Algorithm : https://data-flair.training/blogs/gradient-boosting-algorithm/
- *Christopher Kraussa, Xuan Anh Dob, Nicolas Huckc*,Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500 : https://docs.paperspace.com/machine-learning/wiki/gradient-boosting
- A neural network based approach for predicting customer churn : https://arxiv.org/ftp/arxiv/papers/1309/1309.3945.pdf
- Customer churn : https://blog.hubspot.com/service/what-is-customer-churn
- Prediction of customer churn in e-retailing : https://www.ijrte.org/wp-content/uploads/papers/v8i6/F9550038620.pdf
- Why retail churn is unique : https://www.datamine.com/datafix/index.php/2018/06-why-retail-churn-is-unique-and-how-to-deal-with-it
- Decision Tree : https://www.geeksforgeeks.org/decision-tree/
- Decision Tree : https://scikit-learn.org/stable/modules/tree.html
- How to run a gap analysis for customer service : https://www.custify.com/blog/gap-analysis-for-customer-service/
- Logistic Regression : https://en.wikipedia.org/wiki/Logistic_regression

- Customer churn prediction : https://www.optimove.com/resources/learning-center/customer-churn-prediction-and-prevention
- Understanding Random Forest : https://towardsdatascience.com/understanding-random-forest-58381e0602d2
- Introduction to random forest in Machine Learning : https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/

****