CUSTOMER RETENTION

Submitted by:

JAYASURYA E

Acknowledgment

E-retail factors for customer activation and retention: A case study from Indian e-commerce customers

Customer satisfaction has emerged as one of the most important factors that guarantee the success of online store; it has been posited as a key stimulant of purchase, repurchase intentions and customer loyalty. A comprehensive review of the literature, theories and models have been carried out to propose the models for customer activation and customer retention. Five major factors that contributed to the success of an e-commerce store have been identified as: service quality, system quality, information quality, trust and net benefit. The research furthermore investigated the factors that influence the online customers repeat purchase intention. The combination of both utilitarian value and hedonistic values are needed to affect the repeat purchase intention (loyalty) positively. The data is collected from the Indian online shoppers. Results indicate the e-retail success factors, which are very much critical for customer satisfaction.

Introduction

- ## Business Problem Framing:

Thousands of online purchasing is going on every day. There are some questions every buyer asks himself like: What is the actual retention that this product deserves? Am I purchasing a fair product? In this paper, a machine learning model is proposed to predict a customer retention based on data related to the e-commerce (Age, city, websites etc.). During the development and evaluation of our model, we will show the code used for each step followed by its output. This will facilitate the reproducibility of our work. In this study, Python programming language with a number of Python packages will be used.

- ## Conceptual Background of the Domain Problem:

The main objectives of this study are as follows:

- To apply data pre-processing and preparation techniques in order to obtain clean data
- To build machine learning models able to Customer retention based on data is collected from the Indian online shoppers features.
- To analyse and compare model's performance in order to choose the best model

- # Literature Review

Machine learning is a form of artificial intelligence which compose available computers with the efficiency to be trained without being veraciously programmed. Machine learning interest on the extensions of computer programs which is capable enough to modify when unprotected to new-fangled data. Machine learning algorithms are broadly classified into three divisions, namely; Supervised learning, Unsupervised learning and

Reinforcement learning. Supervised learning is a learning in which we teach or train the machine using data which is well labelled that means some data is already tagged with correct answer. After that, machine is provided with new set of examples so that supervised learning algorithm analyses the training data and produces a correct outcome from labelled data. Unsupervised learning is the training of machine using information that is neither classified nor labelled and allowing the algorithm to act on that information without guidance. Here the task of machine is to group unsorted information according to similarities, patterns and differences without any prior training of data. Unlike, supervised learning, no teacher is provided that means no training will be given to the machine. Therefore, machine is restricted to find the hidden structure in unlabelled data by our-self.

Reinforcement learning is an area of Machine Learning. Reinforcement. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behaviour or path it should take in a specific situation. Reinforcement learning differs from the supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of training dataset, it is bound to learn from its experience. Machine learning has many applications out of which one of the applications is prediction of e-commerce. The e-commerce market is one of the most competitive in terms of retention and same tends to be vary significantly based on lots of factor, forecasting customer retention is an important modules in decision making for both the retailers and customers, finding retention strategies and determining suitable times hence it becomes one of the prime fields to apply the concepts of machine learning to optimize and predict the retention with high accuracy. The study on e-commerce trend is felt important to support the decisions in urban planning. The e-commerce is an unstable stochastic process. Customer's decisions are based on the market trends to reap maximum returns. Developers are interested to know the future trends for their decision making. To accurately estimate Customer retention and future trends, large amount of data that influences the data is collected from the Indian online shoppers is required for analysis, modelling and forecasting. The factors that affect the e-commerce retention have to be studied and their impact on retention has also to be modelled. It is inferred that establishing a simple linear mathematical relationship for these time-series data is found not viable for forecasting. Hence it became imperative to establish a non-linear model which can well fit the data characteristic to analyse and forecast future trends. As the e-commerce is fast developing sector, the analysis and forecast of e-commerce retention using mathematical modelling and other scientific techniques is an immediate urgent need for decision making by all those concerned. The increase in population as well as the industrial activity is attributed to various factors, the most prominent being the recent spurt in the knowledge sector viz. Information Technology (IT) and Information technology enabled services. Demand for e-commerce started of showing an upward trend and the e-commerce activity started booming. Retailer started investing in Indian online shopping Industry. The need for predicting the trend in customer retention was felt by all in the online shopping industry. Therefore, in this paper, we present various important features to use while predicting customer retention with good accuracy. We can use regression models, using various features to have lower Root mean Squared error (RMSE). While using features in a regression model some feature engineering is required for better prediction. Often a set of features linear regression, random forest regression, decision tree regression, xgboost regression and k-nearest neighbors is used for making better model fit. For these models

are expected to be susceptible towards over fitting random forest regression is used to reduce it. So, it directs to the best application of regression models in addition to other techniques to optimize the result.

Linear Regression:

To establish baseline performance with a linear classifier, we used Linear Regression to model the price targets, Y, as a linear function of the data, X

$$f(X) = w_0 + w_1 x_1 + \ \ldots + w_m x_m + x_m$$

$$= \sum_{j=l:m}^{\infty} (w_j x_j)$$

Advantage: A linear model can include more than one predictor as long as the predictors are additive. the best fit line is the line with minimum error from all the points, it has high efficiency but sometimes this high efficiency created.

Disadvantage: Linear Regression Is Limited to Linear Relationships. Linear Regression Only Looks at the Mean of the Dependent Variable. Linear Regression Is Sensitive to Outliers. Data Must Be Independent

Random Forest Regression:

The Random Forest Regression (RFR) is an ensemble algorithm that combines multiple Regression Trees (RTs). Each RT is trained using a random subset of the features, and the output is the average of the individual RTs. The sum of squared errors for a tree T is:

Advantages: There is no need for feature normalization. Individual decision trees can be trained in parallel. Random forests are widely used. They reduce overfitting.

Disadvantages: They're not easily interpretable. They're not a state-of-the-art

$$S = \sum_{c \in leaves(T) i \in C} \sum (y_i - m_c)^2$$

Where $m_c = \frac{1}{n_c} + \sum_{i \in C} y_i$

- # Motivation of the problem undertaken:

Indian online shopping company has decided to check the customer retention on e-commerce sector. The e-commerce sector is decided to use data analytics to know customer retention by their actual values and flip them at a higher retention.

The company is looking at prospective e-commerce to increase customer retentions. You are required to build a model using Machine Learning in order to predict the actual value of the prospective e-commerce and decide the retention of the customer. For this e-commerce sector wants to know:

• Which variables are important to predict the retention of variable?

• How do these variables describe the times of the customer retention?

Analytical Problem Framing

- ## Mathematical/ Analytical Modelling of the Problem Statistical Analysis

  Once it comes time to analyze the data, there are an array of statistical model's analysts may choose to utilize. The most common techniques will fall into the following two groups:

  - Supervised learning, including regression and classification models.
  - Unsupervised learning, including clustering algorithms and association rules

Regression Model:

The regression models are used to examine relationships between variables. Regression models are often used to determine which independent variables hold the most influence over dependent variables information that can be leveraged to make essential decision.
The most traditional regression model is linear regression, decision tree regression, random forest regression, xgboost regression and knn-neighbours.

There are 4 main components of an analytics model, namely: 1) Data Component, 2) Algorithm Component, 3) Real World Component, and 4) Ethical Component.

- # Data Preparation

In this study, we will use a data which is collected from the Indian online shopping company has decided to check the customer retention on e-commerce sector. The e-commerce sector is decided to use data analytics to know customer retention by their actual values and flip them at a higher retention. The data is provided in the optimized_customer_retention.csv file.

- # Data Description

The dataset contains 269 records (rows) and 71 features (columns).

Here, we will provide a brief description of dataset features. Since the number of features is large (71), we will attach the original data description file to this study for more information about the dataset. Now, we will mention the feature name with a short description of its meaning.

'Gender', 'Age', 'City', 'PinCode', 'Years', 'Times', 'Access', 'Device', 'ScSiMobDiv', 'OS', 'Bro
wser', 'Channel', 'Reach', 'Explore', 'PayOpt', 'FreqAbad', 'Abandon', 'ContWeb', 'InfoRev', 'Co
mpInfo', 'RelInfo', 'EaNav', 'LodProSp', 'UsFrInt', 'PayMeth', 'Trust', 'Empathy', 'GuarPriv', 'R
esAvab', 'BenfDisc', 'Enjoyment', 'ConvFlex', 'RetRepPol', 'GaAcLoPr', 'DispQualinfo', 'SatSh

GoQual', 'NetBenf', 'SatExTru', 'WidVarProd', 'CompRelProdInfo', 'MontSav', 'ConvPat', 'Sen Adv', 'ETSocStat', 'GratFavET', 'FulCerRol', 'ValMonSp', 'OnRetShop', 'EaUsWeb', 'VisApp WebLay', 'WiVaProdOff', 'CompRelProd', 'FaLoWebSp', 'RelWeb', 'QuComPur', 'AvaSevPay Opt', 'SpOrdDel', 'PrCusInfo', 'SecFinInfo', 'PerTrWor', 'PreAsMulCh', 'LoTiLogIn', 'LoTiDis GrPh', 'LaDecPr', 'LonPaLoTi', 'LimMoPayPr', 'LoDelPer', 'ChAppDes', 'FrqDist', 'WebEffBe f', 'IndOnlRet'

- # Data Pre-processing:

Unique Function for dataset:

There are some column features which is indicated with both integers and string values so we are replacing such feature into unique functions. They are age, years, times, screen size of mobile, explore.

Then we move to see statistical information about the non-numerical columns in our dataset:

```
# Stastistical summary
df_describe=df.describe()
df_describe
```

	Gender	Age	City	PinCode	Years	Times	Access	Device	ScSiMobDiv	OS	Browser	Channel
count	269.000000	269.000000	269.000000	269.000000	269.000000	269.000000	269.000000	269.000000	269.000000	269.000000	269.000000	269.000000
mean	0.327138	1.330855	4.721190	220994.423792	2.524164	1.695167	1.267658	1.501859	1.263941	1.137546	0.524164	1.780669
std	0.470042	1.183774	3.118367	141810.976009	1.436586	1.689524	0.476025	0.751240	1.390614	0.867985	1.097985	0.566672
min	0.000000	0.000000	0.000000	110000.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	2.000000	122000.000000	2.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.000000	2.000000
50%	0.000000	1.000000	5.000000	201300.000000	3.000000	1.000000	1.000000	2.000000	1.000000	1.000000	0.000000	2.000000
75%	1.000000	2.000000	8.000000	201300.000000	4.000000	3.000000	2.000000	2.000000	3.000000	2.000000	0.000000	2.000000
max	1.000000	4.000000	9.000000	560000.000000	4.000000	5.000000	2.000000	3.000000	3.000000	2.000000	3.000000	2.000000

From the table above, we can see, for example, that the mean times of the customer retentions in our dataset is 0.32 with a standard deviation of 0.47. We can see also that the minimum times is 0 and the maximum times is 5 with a median of 1. Similarly, we can get a lot of information about our dataset variables from the table.

There are no null values in the dataset. Since, we can see lot of string values in the dataset, so we proceed with further steps.

- # Data Cleaning:

Column	Data types	Column	Data types
Gender	object	NetBenf	int64
Age	object	SatExTru	int64
City	object	WidVarProd	int64
PinCode	int64	CompRelProdInfo	int64
Years	object	MontSav	int64
Times	object	ConvPat	int64
Access	object	SenAdv	int64
Device	object	ETSocStat	int64
ScSiMobDiv	object	GratFavET	int64
OS	object	FulCerRol	int64
Browser	object	ValMonSp	int64

Channel	object	OnRetShop	object
Reach	object	EaUsWeb	object
Explore	object	VisAppWebLay	object
PayOpt	object	WiVaProdOff	object
FreqAbad	object	CompRelProd	object
Abandon	object	FaLoWebSp	object
ContWeb	int64	RelWeb	object
InfoRev	int64	QuComPur	object
CompInfo	int64	AvaSevPayOpt	object
RelInfo	int64	SpOrdDel	object
EaNav	int64	PrCusInfo	object
LodProSp	int64	SecFinInfo	object
UsFrInt	int64	PerTrWor	object
PayMeth	int64	PreAsMulCh	object
Trust	int64	LoTiLogIn	object
Empathy	int64	LoTiDisGrPh	object
GuarPriv	int64	LaDecPr	object
ResAvab	int64	LonPaLoTi	object
BenfDisc	int64	LimMoPayPr	object
Enjoyment	int64	LoDelPer	object
ConvFlex	int64	ChAppDes	object
RetRepPol	int64	FrqDist	object
GaAcLoPr	int64	WebEffBef	object
DispQualinfo	int64	IndOnlRet	object
SatShGoQual	int64		

Now, we can proceed with encoding techniques to convert the string data to numerical one.

- **Encoding of Data Frame:**

In ordinal encoding, each unique category value is assigned an integer value. This ordinal encoding transform is available in the scikit-learn Python machine learning library via the Ordinal Encoder class. By default, it will assign integers to labels in the order that is observed in the data.

In this encoding scheme, the categorical feature is first converted into numerical using an ordinal encoder. Then the numbers are transformed in the binary number. After that binary value is split into different columns. Binary encoding works really well when there are a high number of categories.

```
from sklearn.preprocessing import OrdinalEncoder
enc=OrdinalEncoder()
```

```
for i in df.columns:
    if df[i].dtypes=="object":
        df[i]=enc.fit_transform(df[i].values.reshape(-1,1))
```

```
df.head()    # informtion about top of the data after Ordinal encoder
```

	Gender	Age	City	PinCode	Years	Times	Access	Device	ScSiMobDiv	OS	Browser	Channel	Reach	Explore	PayOpt	FreqAbad	Abandon	ContWeb
0	1.0	1.0	2.0	110000	4.0	3.0	0.0	0.0	0.0	2.0	0.0	2.0	2.0	4.0	2.0	2.0	2.0	4
1	0.0	0.0	2.0	110000	4.0	4.0	2.0	2.0	1.0	1.0	0.0	2.0	4.0	2.0	1.0	3.0	4.0	5
2	0.0	0.0	8.0	201300	3.0	4.0	1.0	2.0	3.0	0.0	0.0	2.0	4.0	1.0	2.0	2.0	4.0	5
3	1.0	0.0	5.0	132000	3.0	0.0	1.0	2.0	3.0	1.0	3.0	2.0	2.0	4.0	1.0	1.0	0.0	4
4	0.0	0.0	0.0	560000	2.0	1.0	2.0	2.0	1.0	1.0	3.0	0.0	4.0	2.0	1.0	0.0	0.0	5

- **Correlation matrix:**

A correlation matrix is simply a table which displays the correlation. The measure is best used in variables that demonstrate a linear relationship between each other. The fit of the data can be visually represented in a heatmap.

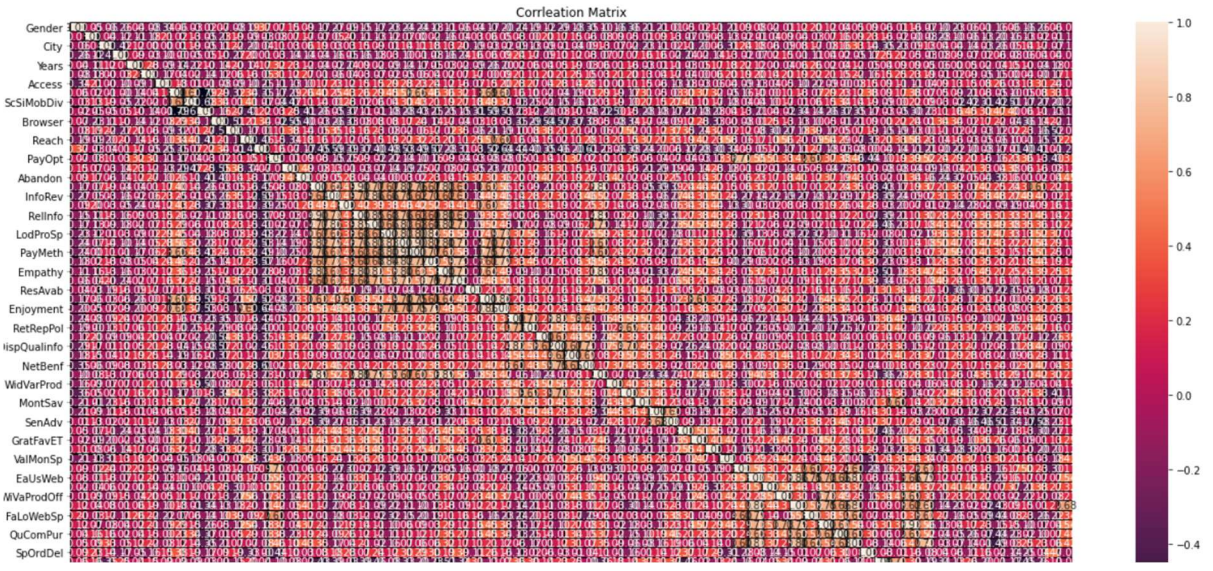
Pandas dataframe. corr() method is used for creating the correlation matrix. It is used to find the pairwise correlation of all columns in the dataframe.

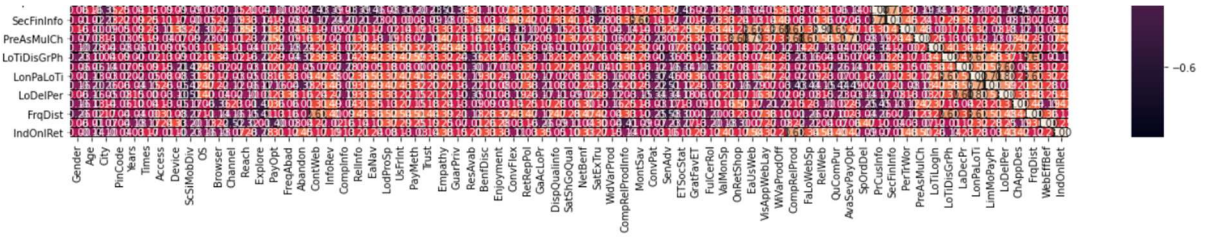
To create correlation matrix using pandas, these steps should be taken:

1. Obtain the data.
2. Create the DataFrame using Pandas.
3. Create correaltion matrix using Pandas.

```
corr_mat=df.corr()
corr_mat
```

	Gender	Age	City	PinCode	Years	Times	Access	Device	ScSiMobDiv	OS	Browser	Channel
Gender	1.000000	0.046169	-0.062279	-0.260679	0.037982	-0.080696	-0.342762	0.061673	-0.029837	-0.019243	0.071385	-0.079834
Age	0.046169	1.000000	0.035190	-0.124064	-0.113329	-0.176995	0.206453	0.022383	0.134888	-0.048087	-0.231530	0.186455
City	-0.062279	0.035190	1.000000	-0.415184	0.116037	0.002222	0.002701	0.012168	0.187405	-0.053328	-0.108638	0.222877
PinCode	-0.260679	-0.124064	-0.415184	1.000000	0.090686	-0.028720	-0.095907	-0.003308	-0.053663	-0.053171	0.095108	-0.271433
Years	0.037982	-0.113329	0.116037	0.090686	1.000000	0.275155	-0.085882	-0.144390	-0.215198	0.103555	-0.139349	0.196752
Times	-0.080696	-0.176995	0.002222	-0.028720	0.275155	1.000000	-0.171903	0.038666	0.015315	-0.141778	-0.124746	0.058518
Access	-0.342762	0.206453	0.002701	-0.095907	-0.085882	-0.171903	1.000000	0.175984	-0.011294	0.009904	-0.076672	-0.085875
Device	0.061673	0.022383	0.012168	-0.003308	-0.144390	0.038666	0.175984	1.000000	0.622797	-0.787216	0.290590	-0.318963
ScSiMobDiv	-0.029837	0.134888	0.187405	-0.053663	-0.215198	0.015315	-0.011294	0.622797	1.000000	-0.676279	0.336715	0.012181
OS	-0.019243	-0.048087	-0.053328	-0.053171	0.103555	-0.141778	0.009904	-0.787216	-0.676279	1.000000	-0.158152	0.266389





Observations: We are unable to identify the correlation in above heatmap due to huge number of columns.

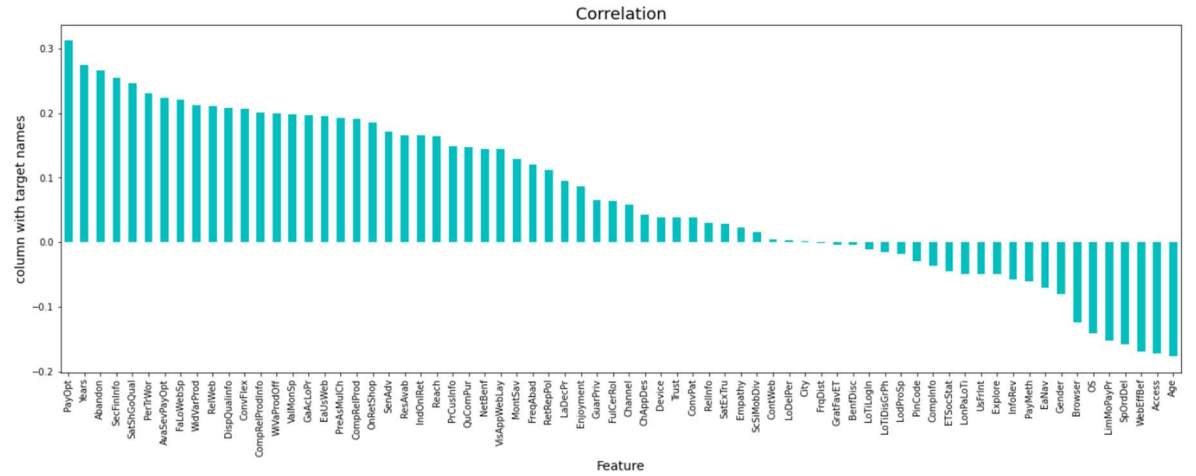
How correlation matrix is calculated?

A correlation matrix is a table showing correlation coefficients between sets of variables. Each random variable (x) in the table is correlated with each of the other values in the table x. The diagonal of the table is always a set of ones, because the correlation between a variable and itself is always 1.

Column	Data types	Column	Data types
Times	1.000000	Channel	0.058518
PayOpt	0.312125	ChAppDes	0.042634
Years	0.275155	Device	0.038666
Abandon	0.266509	Trust	0.038543
SecFinInfo	0.254985	ConvPat	0.038159
SatShGoQual	0.246457	RelInfo	0.030285
PerTrWor	0.230943	SatExTru	0.028708
AvaSevPayOpt	0.224067	Empathy	0.022695
FaLoWebSp	0.220599	ScSiMobDiv	0.015315
WidVarProd	0.212938	ContWeb	0.005060
RelWeb	0.211079	LoDelPer	0.003016
DispQualinfo	0.208380	City	0.002222
ConvFlex	0.206120	FrqDist	-0.001586
CompRelProdInfo	0.200574	GratFavET	-0.003574
WiVaProdOff	0.200184	BenfDisc	-0.004580
ValMonSp	0.198845	LoTiLogIn	-0.010795
GaAcLoPr	0.197486	LoTiDisGrPh	-0.015164
EaUsWeb	0.195441	LodProSp	-0.017747
PreAsMulCh	0.193081	PinCode	-0.028720
CompRelProd	0.190751	CompInfo	-0.035907
OnRetShop	0.185847	ETSocStat	-0.044747
SenAdv	0.170777	LonPaLoTi	-0.048663
ResAvab	0.165773	UsFrInt	-0.048798
IndOnlRet	0.165071	Explore	-0.049347
Reach	0.164312	InfoRev	-0.057061
PrCusInfo	0.148304	PayMeth	-0.059943
QuComPur	0.146791	EaNav	-0.069875
NetBenf	0.145172	Gender	-0.080696
VisAppWebLay	0.144759	Browser	-0.124746
MontSav	0.129452	OS	-0.141778
FreqAbad	0.120437	LimMoPayPr	-0.152589
RetRepPol	0.112018	SpOrdDel	-0.158309
LaDecPr	0.094715	WebEffBef	-0.168893
Enjoyment	0.086905	Access	-0.171903
GuarPriv	0.065069	Age	-0.176995
FulCerRol	0.064003		

Now we can clearly identify the correlation of independent variables with the target variables "Times". There are around 27 variables who has less than 0.01 correlation value (very week relationship.)

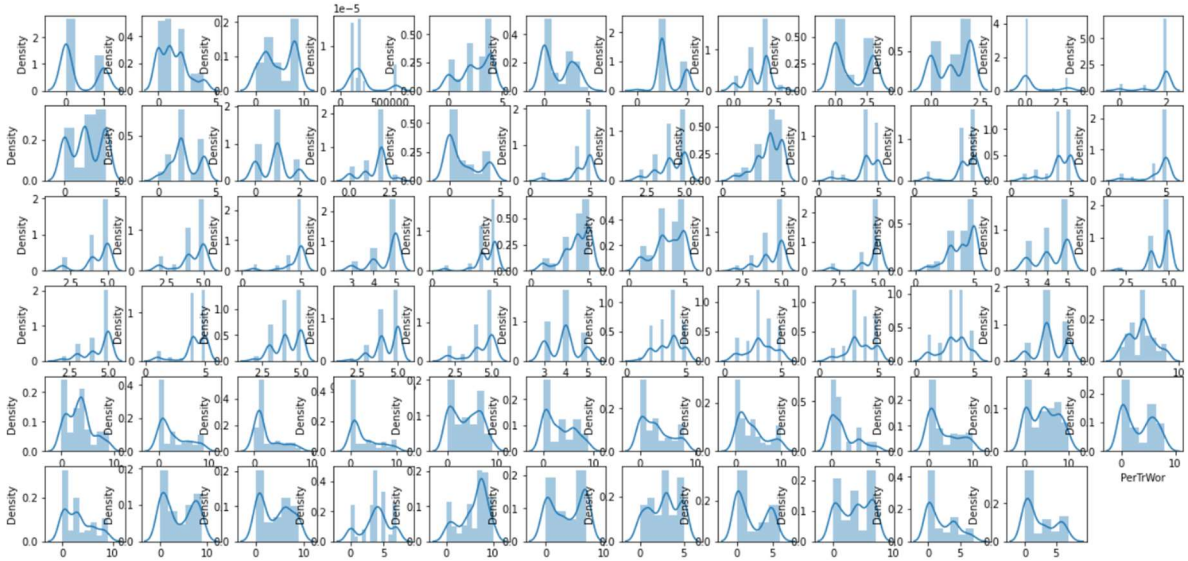
Checking the columns which are positively and negative correlated with the target columns:



Outcome of Correlation:

The Columns of the dataset is Correlated in both Positively and Negatively with target columns.
The Positive and negative correlation values is shown in both numbers and graph.
Max correlation: Payment Option (PayOpt)
Min correlation: Age

Let's check the data distribution among all the columns.

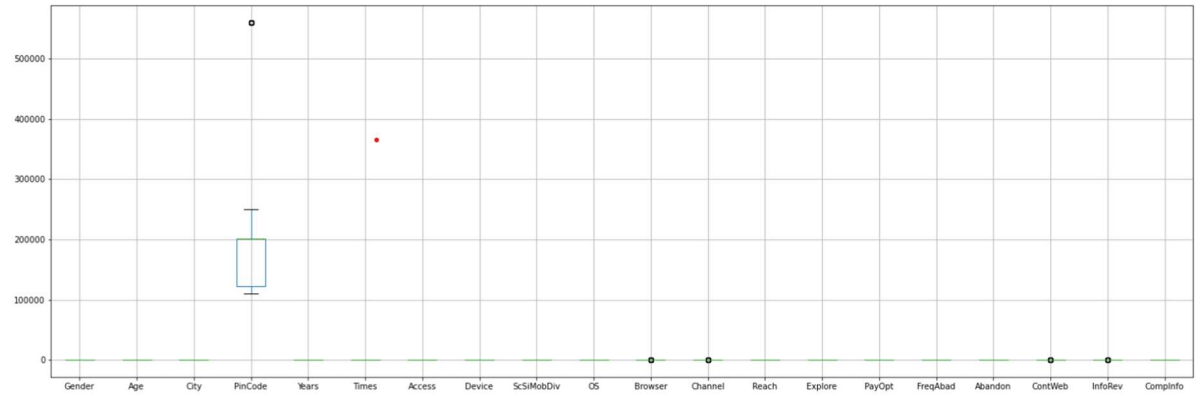


We can see skewness in data for the multiple columns, will handle the skewness in further steps.

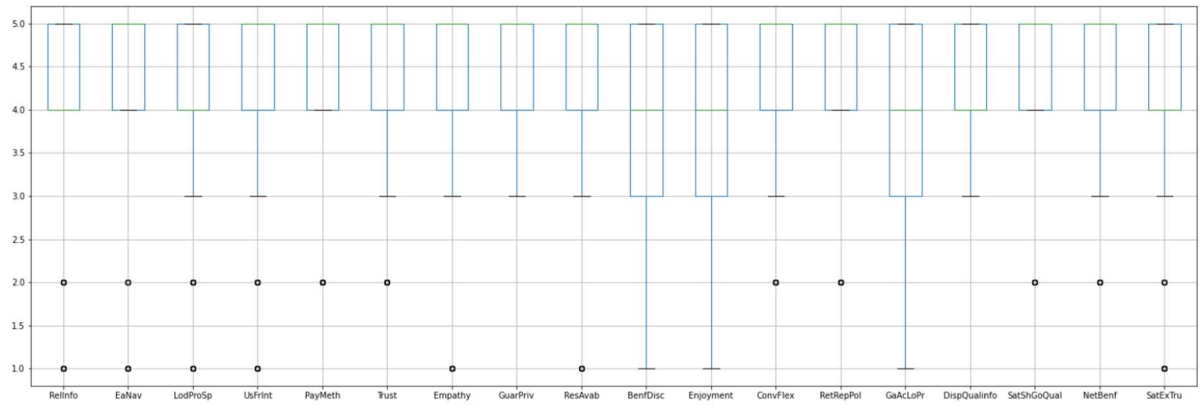
- **Outliers Check:**

There are 71 columns in dataset so it's not possible to plot each and every column separately or plot all together. so, we will print in 4 steps:

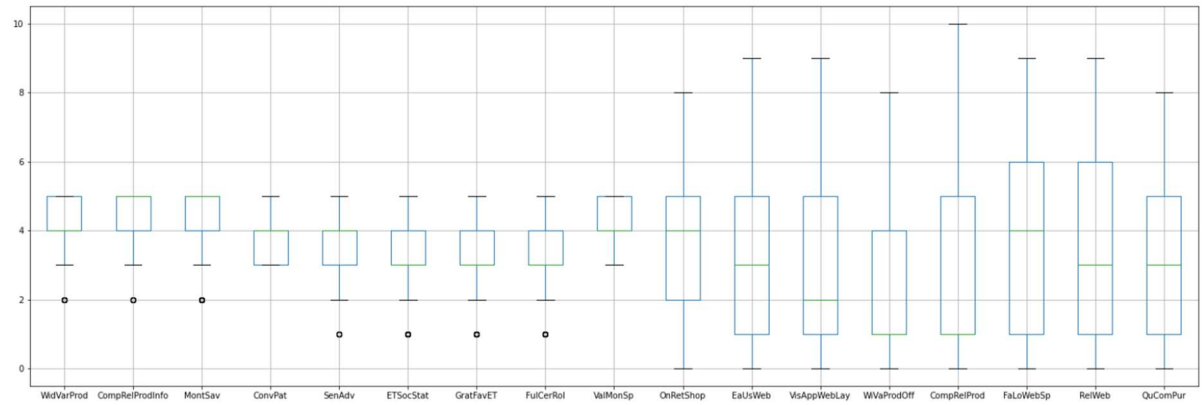
```
#Plotting Boxplots for first 20 columns
df.iloc[:,0:20].boxplot(figsize=[25,10])
plt.subplots_adjust(bottom=0.25)
plt.show()
```



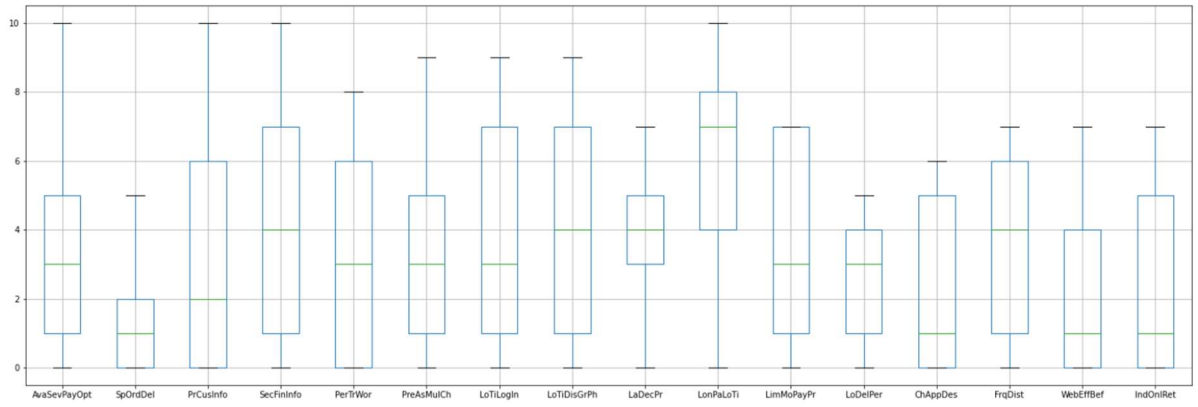

```
#Plotting boxplot for next 18 columns
df.iloc[:,20:38].boxplot(figsize=[25,10])
plt.subplots_adjust(bottom=0.25)
plt.show()
```




```
#Plotting boxplot for next 18 columns
df.iloc[:,38:55].boxplot(figsize=[25,10])
plt.subplots_adjust(bottom=0.25)
plt.show()
```



```
#Plotting boxplot for remaining columns
df.iloc[:,55:71].boxplot(figsize=[25,10])
plt.subplots_adjust(bottom=0.25)
plt.show()
```



Now let's see outliers in few features so we consider z-score technique to remove the outliers.

Removal of Outliers:

Let us consider Z-score technique, it removes the outliers in the dataset.

Finally, the percentage of data loss in dataset is 25.65 due to larger number of outliers.

- # Skewness:

Skewness is a measure of symmetry in a distribution. Actually, it's more correct to describe it as a measure of lack of symmetry. A standard normal distribution is perfectly symmetrical and has zero skew. Therefore, we need a way to calculate how much the distribution is skewed.

- # Checking Skewness:

Columns	Skewness	Columns	Skewness
Gender	0.741028	NetBenf	-1.180598
Age	0.680987	SatExTru	-1.814791
City	-0.105049	WidVarProd	-0.551863
PinCode	1.751959	CompRelProdInfo	-0.940104
Years	-0.554705	MontSav	-1.247623
Times	0.286241	ConvPat	0.115519
Access	0.630053	SenAdv	-0.337315
Device	-0.485180	ETSocStat	-0.248103
ScSiMobDiv	0.349042	GratFavET	-0.360913
OS	-0.270283	FulCerRol	-0.495053
Browser	1.710244	ValMonSp	-0.172501
Channel	-2.469485	OnRetShop	0.145462
Reach	-0.041161	EaUsWeb	0.508414
Explore	0.155075	VisAppWebLay	0.813015
PayOpt	0.129735	WiVaProdOff	1.186071
FreqAbad	-0.867449	CompRelProd	0.964100
Abandon	0.743052	FaLoWebSp	0.106301
ContWeb	-2.234127	RelWeb	0.464836
InfoRev	-0.801079	QuComPur	0.606858
CompInfo	-0.893446	AvaSevPayOpt	0.781633
RelInfo	-1.710957	SpOrdDel	1.061243

EaNav	-2.052560	PrCusInfo	0.716860
LodProSp	-1.467621	SecFinInfo	0.073523
UsFrInt	-2.015996	PerTrWor	0.199441
PayMeth	-1.581400	PreAsMulCh	0.577166
Trust	-1.261484	LoTiLogIn	0.131096
Empathy	-2.294982	LoTiDisGrPh	0.167550
GuarPriv	-1.355737	LaDecPr	-0.378929
ResAvab	-2.104016	LonPaLoTi	-0.708594
BenfDisc	-1.052475	LimMoPayPr	-0.086712
Enjoyment	-0.565041	LoDelPer	-0.147702
ConvFlex	-1.121619	ChAppDes	0.354163
RetRepPol	-2.243625	FrqDist	-0.100608
GaAcLoPr	-0.853530	WebEffBef	0.662084
DispQualinfo	-0.555681	IndOnlRet	0.583614
SatShGoQual	-1.989886		

To handle skewness of the data using different types of functions:

1. Log Transform
2. Square Root Transform
3. Box-Cox Transform
4. Power transform

Now here, we are going to use Power transform function to handle skewness in dataset.

Then, splitting the independent and target variable in x and y.

In statistics, a power transform is a family of functions applied to create a monotonic transformation of data using power functions. It is a data transformation technique used to stabilize variance, make the data more normal distribution-like, improve the validity of measures of association (such as the Pearson correlation between variables), and for other data stabilization procedures.

```
from sklearn.preprocessing import power_transform
df_new=power_transform(x)
df_new=pd.DataFrame(df_new,columns=x.columns)
```

After performing such statistics, the skewness is removed in dataset as shown below:

Columns	**Skewness**	**Columns**	**Skewness**
Gender	0.741028	NetBenf	-0.622457
Age	-0.024246	SatExTru	-0.469379
City	-0.263529	WidVarProd	-0.205770
PinCode	0.000000	CompRelProdInfo	-0.317747
Years	-0.371550	MontSav	-0.553624
Access	0.006801	ConvPat	-0.018599
Device	-0.138222	SenAdv	-0.136339
ScSiMobDiv	0.165368	ETSocStat	-0.114445
OS	-0.272132	GratFavET	-0.108012
Browser	1.533776	FulCerRol	-0.152034
Channel	-2.029216	ValMonSp	-0.048997
Reach	-0.208762	OnRetShop	-0.115650

Explore	-0.055691	EaUsWeb	-0.084097
PayOpt	-0.083889	VisAppWebLay	0.006539
FreqAbad	-0.192075	WiVaProdOff	0.056626
Abandon	0.231205	CompRelProd	0.042060
ContWeb	-0.770947	FaLoWebSp	-0.203340
InfoRev	-0.281012	RelWeb	-0.083203
CompInfo	-0.235305	QuComPur	-0.064882
RelInfo	-0.406237	AvaSevPayOpt	-0.025756
EaNav	-0.588561	SpOrdDel	0.130552
LodProSp	-0.411050	PrCusInfo	0.010433
UsFrInt	-1.110510	SecFinInfo	-0.233972
PayMeth	-0.647029	PerTrWor	-0.140989
Trust	-0.495785	PreAsMulCh	-0.082562
Empathy	-1.152407	LoTiLogIn	-0.169011
GuarPriv	-0.848203	LoTiDisGrPh	-0.164309
ResAvab	-0.539238	LaDecPr	-0.242822
BenfDisc	-0.341753	LonPaLoTi	-0.406803
Enjoyment	-0.242151	LimMoPayPr	-0.265289
ConvFlex	-0.453152	LoDelPer	-0.226723
RetRepPol	-1.132389	ChAppDes	-0.002355
GaAcLoPr	-0.314912	FrqDist	-0.271126
DispQualinfo	-0.297591	WebEffBef	0.018153
SatShGoQual	-0.676094	IndOnlRet	0.006779

- # Hardware and Software Requirements and Tools Used

PYTHON Jupyter Notebook:

Key Features:

An open-source solution that has simple coding processes and syntax so it's fairly easy to learn Integration with other languages such as C/C++, Java, PHP, C#, etc.
Advanced analysis processes through machine learning and text mining.

Python is extremely accessible to code in comparison to other popular languages such as Java, and its syntax is relatively easy to learn making this tool popular among users that look for an open-source solution and simple coding processes. In data analysis, Python is used for data crawling, cleaning, modelling, and constructing analysis algorithms based on business scenarios. One of the best features is actually its user-friendliness: programmers don't need to remember the architecture of the system nor handle the memory – Python is considered a high-level language that is not subject to the computer's local processor.

Libraries and Packages used:

Matplotlib:

Matplotlib is a Python library that uses Python Script to write 2-dimensional graphs and plots. Often mathematical or scientific applications require more than single axes in a representation. This library helps us to build multiple plots at a time. You can, however, use Matplotlib to manipulate different characteristics of figures as well.

The task carried out is visualization of dataset i.e., nominal data, ordinal data, continuous data, heatmap display distribution for correlation matrix and null values, boxplot distribution for checking outliers, scatter plot distribution for modelling approach, subplot distribution for analysis and comparison, feature importance and common importance features, line plot for prediction values vs actual values.

Numpy:

Numpy is a popular array – processing package of Python. It provides good support for different dimensional array objects as well as for matrices. Numpy is not only confined to providing arrays only, but it also provides a variety of tools to manage these arrays. It is fast, efficient, and really good for managing matrices and arrays.

The Numpy is used to managing matrices i.e., MAE, MSE and RMSE and arrays i.e., described the values of train test dataset.

Pandas:

Pandas is a python software package. It is a must to learn for data-science and dedicatedly written for Python language. It is a fast, demonstrative, and adjustable platform that offers intuitive data-structures. You can easily manipulate any type of data such as – structured or time-series data with this amazing package.

The Pandas is used to execute a Data frame i.e., test set.csv, train set.csv, skewness, co-efficient, predicted values of model approach, conclusion.

Scikit Learn:

Scikit learn is a simple and useful python machine learning library. It is written in python, cython, C, and C++. However, most of it is written in the Python programming language. It is a free machine learning library. It is a flexible python package that can work in complete harmony with other python libraries and packages such as Numpy and Scipy.

Scikit learn library is used to import a pre-processing function i.e., power transform, ordinal encoder, minimax scaler, linear, random forest, decision tree, xgboost, k-nearest neighbours, r2 score, mean absolute error, mean squared error, train test split, grid search cv and ensemble technique.

Models Development and Evaluation

In this section, we choose the type of machine learning prediction that is suitable to our problem. We want to determine if this is a regression problem or a classification problem. In this project, we want to predict the times of a customer retention given information about it. The times we want to predict is a continuous value; it can be any real number. This can be seen by looking at the target variable in our dataset Times:

```
df['Times'].value_counts()     # Checking the value counts of Target variable.
```

```
Less than 10 times    114
31-40 times            63
41 times and above     47
11-20 times            29
21-30 times            10
42 times and above      6
```

That means that the prediction type that is appropriate to our problem is regression. Now, we move to choose the modelling techniques we want to use. There are a lot of techniques available for regression problems like Linear Regression, Decision Trees, Random Forest, XGBoost, k-nearest neighbors (KNN) etc. In this project, we will test many modelling techniques, and then choose the technique(s) that yield the best results. The techniques that we will try are:

1. Linear Regression

This technique models the relationship between the target variable and the independent variables (predictors). It fits a linear model with coefficients to the data in order to minimize the residual sum of squares between the target variable in the dataset, and the predicted values by the linear approximation.

2. Random Forest

Bagging is an ensemble method where many base models are used with a randomized subset of data to reduce the variance of the base model.

3. Decision Trees

For this technique, the goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

Each one of these techniques has many algorithmic implementations. We will choose algorithm(s) for each of these techniques in the next section.

4. XGBoost

It's a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks.

5. k-nearest neighbors (KNN)

The k-nearest neighbors (KNN) algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems.

Model Building and Evaluation

In this part, we will build our prediction model: we will choose algorithms for each of the techniques we mentioned in the previous section. After we build the model, we will evaluate its performance and results.

Feature Scaling:

In order to make all algorithms work properly with our data, A way to normalize the input features/variables is the Min-Max scaler. By doing so, all features will be transformed into the range [0,1] meaning that the minimum and maximum value of a feature/variable is going to be 0 and 1, respectively.

Importing libraries for metrics and model building:

```
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from xgboost import XGBRegressor
from sklearn.neighbors import KNeighborsRegressor
```

Modelling Approach:

For each one of the techniques mentioned in the previous section (Linear Regression, Random Forest Regression, Decision Tree Regression, XGBoost, k-nearest neighbors (KNN) etc etc.), we will follow these steps to build a model:

- Choose an algorithm that implements the corresponding technique
- Search for an effective parameter combination for the chosen algorithm
- Create a model using the found parameters
- Train (fit) the model on the training dataset
- Test the model on the test dataset and get the results

Regression Method:

- Using Scikit-Learn, we can build a model.

```
maxR2=0
BestRS=0
for i in range(1,200):
    x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=.25,random_state=i)
    LR = LinearRegression()
    LR.fit(x_train, y_train)
    predrf = LR.predict(x_test)
    r2=r2_score(y_test, predrf)
    if r2>maxR2:
        maxR2=r2
        BestRS=i
print("Best R2 is " ,maxR2," on Random_state ",BestRS)
```

```
Best R2 is  0.30209556500533163  on Random_state  43
```

Splitting the Dataset:

As usual for supervised machine learning problems, we need a training dataset to train our model and a test dataset to evaluate the model. So, we will split our dataset randomly into

two parts, one for training and the other for testing. For that, we will use another function from Scikit-Learn called train_test_split():

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size = 0.25,random_state = BestRS)
```

Performance Metric:

For evaluating the performance of our models, we will use R2 score, mean absolute error (MAE) and mean squared error (MSE). If the predicted value of the element, and the corresponding true value, then for all the elements, RMSE is calculated as:

```
def eval(x):
    mod =x
    mod.fit(x_train, y_train)
    predict_test = mod.predict(x_test)
    print("R2 score is ", r2_score(y_test, predict_test)*100)
    print("Mean Absolute error is", mean_absolute_error(y_test,predict_test))
    print("Mean squared error is", mean_squared_error(y_test,predict_test))
    print("Root mean squared error is", np.sqrt(mean_squared_error(y_test,predict_test)))
```

Model Building	R2 score	MAE	MSE	RMSE
Linear	30.20	1.17	1.91	1.38
Random	72.23	0.42	0.76	0.87
Decision	77.94	0.21	0.60	0.77
XGBoost	71.51	0.28	0.78	0.88
KNeighbors	33.47	0.98	1.82	1.35

Comparing the Performance metric and Cross validation Score:

Performance Metric	Cross -Validation Score
30.20	-1.72
72.23	77.91
77.94	75.97
71.51	76.12
33.47	42.58

Here we have handled the problem of the overfitting and the underfitting by checking the R2 score.
Comparing the performance metric and cross-validation score which has minimum difference is decision tree. so finally, this is our best model.

Hyper Parameter Tuning:

Hyperparameters are crucial as they control the overall behaviour of a machine learning model. The ultimate goal is to find an optimal combination of hyperparameters that minimizes a predefined loss function to give better results.

The performance metric for Decision Tree Regressor has higher values when compared with other values.so, we are going to perform hyperparameter tuning for this model to get better result.

Importing GridSearchCV

```
from sklearn.model_selection import GridSearchCV
```

Hyperparameter Tuning for Decision Tree Regressor:

Firstly, we will use GridSearchCV() to search for the best model parameters in a parameter space provided by us. criterion, max features and random state.

```
from sklearn.tree import DecisionTreeRegressor
parameters={'criterion' : ['mse', 'friedman_mse', 'mae', 'poisson'], 'max_features' : ['auto','sqrt','log2'],
            'splitter' : ['best', 'random'], 'random_state' : list(range(0,10))}
dtr=DecisionTreeRegressor()
clf=GridSearchCV(dtr,parameters)
clf.fit(x_train,y_train)
print(clf.best_params_)
```

```
{'criterion': 'mae', 'max_features': 'auto', 'random_state': 0, 'splitter': 'best'}
```

We defined the parameter space above using reasonable values for chosen parameters.

```
dtr=DecisionTreeRegressor(criterion = 'mae', max_features = 'auto', random_state = 0, splitter = 'best')
dtr.fit(x_train,y_train)
dtr.score(x_train,y_train)
pred_decision=dtr.predict(x_test)
dtrs=r2_score(y_test,pred_decision)
print('R2 Score:',dtrs*100)
dtrscore=cross_val_score(dtr,x,y,cv=5)
dtrc=dtrscore.mean()
print('Cross Val Score:',dtrc*100)
```

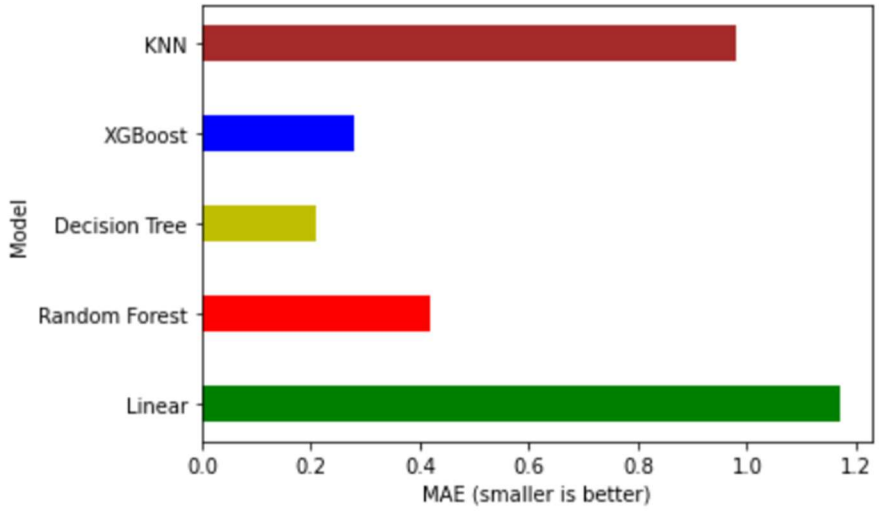
```
R2 Score: 88.23529411764706
Cross Val Score: 71.81426309167811
```

We defined the performance model score and cross validation score of hyperparameter tuning for decision tree using chosen parameters. We are getting model accuracy and cross validation has 88.23% & 71.81% respectively. We consider Decision Tree regressor is our best model for these datasets.

Performance Interpretation:

MAE (Mean Absolute Error):

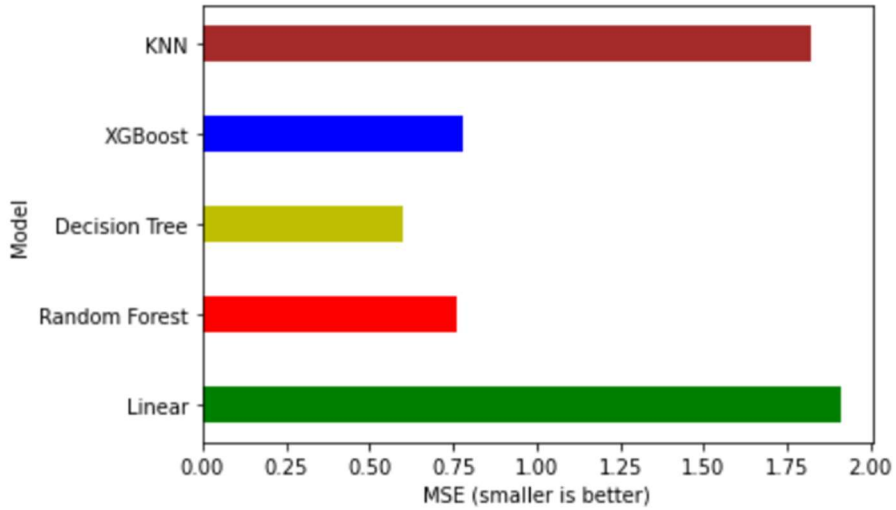
```
x = ['Linear','Random Forest', 'Decision Tree', 'XGBoost', 'KNN']
y = [1.17, 0.42 , 0.21, 0.28, 0.98]
colors = ["g", "r", "y", "b", "brown"]
fig, js = plt.subplots()
plt.barh(y=range(len(x)), tick_label=x, width=y, height=0.4, color=colors);
js.set(xlabel="MAE (smaller is better)", ylabel="Model");
```



By looking at the table and the graph, we can see that Decision Tree has the smallest MSE, 0.60. After that, Random Forest and XGBoost comes with similar errors: 0.76 and 0.78 respectively. At last, the K-Nearest Neighbors and Linear comes with an similar errors: 1.82 and 1.91 respectively.
So, in our experiment, the best model is Decision Tree and the worst model is Linear. We can see that the difference in MSE between the best model and the worst model is significant; the best model has least error of the worst model.

MSE (Mean Squared Error) :

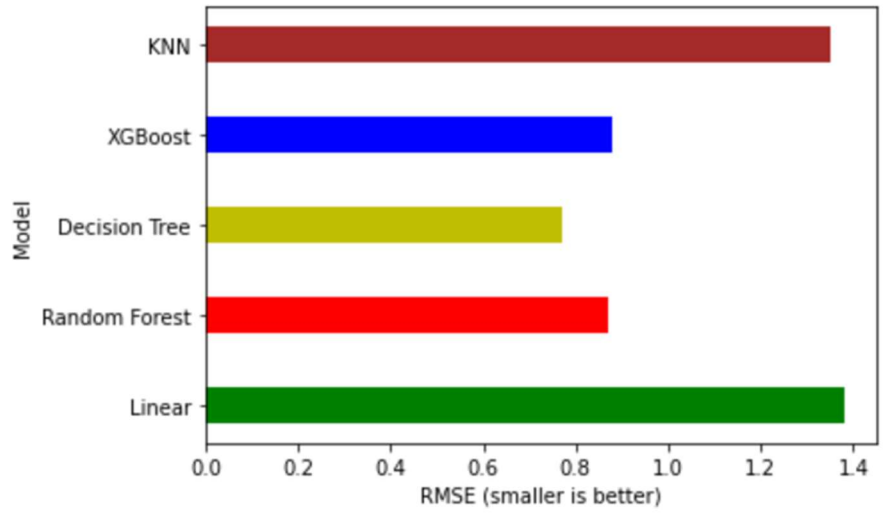
```
x = ['Linear','Random Forest', 'Decision Tree', 'XGBoost', 'KNN']
y = [1.91, 0.76, 0.60, 0.78, 1.82]
colors = ["g", "r", "y", "b", "brown"]
fig, js = plt.subplots()
plt.barh(y=range(len(x)), tick_label=x, width=y, height=0.4, color=colors);
js.set(xlabel="MSE (smaller is better)", ylabel="Model");
```



By looking at the table and the graph, we can see that Decision Tree has the smallest MSE, 0.60. After that, Random Forest and XGBoost comes with similar errors: 0.76 and 0.78 respectively. At last, the K-Nearest Neighbors and Linear comes with an similar errors: 1.82 and 1.91 respectively.
So, in our experiment, the best model is Decision Tree and the worst model is Linear. We can see that the difference in MSE between the best model and the worst model is significant; the best model has least error of the worst model.

RMSE (Root Mean Squared Error)

```
x = ['Linear','Random Forest', 'Decision Tree','XGBoost', 'KNN']
y = [1.38, 0.87, 0.77, 0.88, 1.35]
colors = ["g", "r", "y", "b", "brown"]
fig, js = plt.subplots()
plt.barh(y=range(len(x)), tick_label=x, width=y, height=0.4, color=colors);
js.set(xlabel="RMSE (smaller is better)", ylabel="Model");
```



By looking at the table and the graph, we can see that Decision Tree has the smallest RMSE of 0.77. After that, Random Forest and XGBoost comes with similar errors: 0.87 and 0.88 respectively. At last, the K-Nearest Neighbors and linear comes with a similar error: 1.35 and 1.38 respectively.

So, in our experiment, the best model is Decision Tree and the worst model is Linear. We can see that the difference in RMSE between the best model and the worst model is significant; the best model has almost least error of the worst model.

We chose the root mean squared error (RMSE) as our performance metric to evaluate and co mpare models. RMSE presents a value that is easy to understand; it shows the average value of model error. For example, for our Decision Tree model, its RMSE is 0.77, which means th at on average Decision Tree will predict a value that is bigger or smaller than the true value b y 0.77. Now to understand how good this RMSE is, we need to know the range and distributi on of the data.
Finally, we came to know that our best model is Decision Tree, then the worst model is Linear.

Feature Importance's:

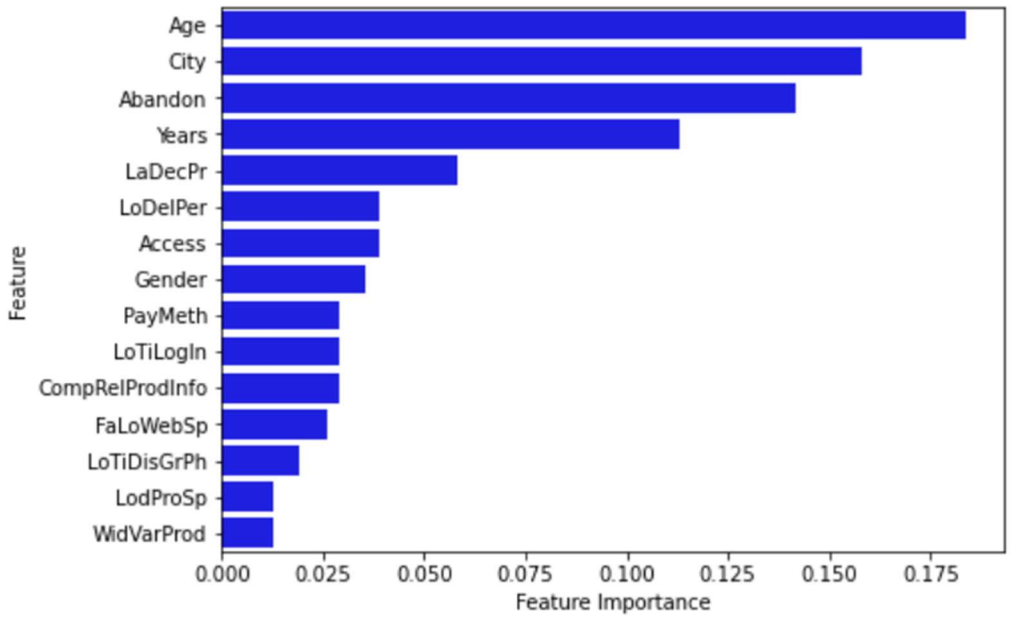
Some of the models we used provide the ability to see the importance of each feature in the dataset after fitting the model. We will look at the feature importance's provided by Decision Tree models. We have 71 features in our data which is a big number, so we will take a look at the 15 most important features.

Decision Tree

Now, let's see the most important features as for Decision Tree model:

```
dtr_feature_importances = dtr.feature_importances_
dtr_feature_importances = pd.Series(dtr_feature_importances,
                                    index=x_train.columns.values).sort_values(ascending=False).head(15)

fig, js = plt.subplots(figsize=(7,5))
sns.barplot(x=dtr_feature_importances, y=dtr_feature_importances.index, color="b");
plt.xlabel('Feature Importance');
plt.ylabel('Feature');
```



Notice here in feature importance of Random Forest, the age feature plays a prominent role for target variable.

Conclusion:

In this paper, we built several regression models to predict the times of some customer retention given some of the customer retention features. We evaluated and compared each model to determine the one with highest performance. We also looked at how some models rank the features according to their importance. In this paper, we followed the data science process starting with getting the data, then cleaning and pre-processing the data, followed by exploring the data and building models, then evaluating the results.

As a recommendation, we advise to use this model (or a version of it trained with more recent data) by e-commerce sector who want to get an idea about customer. The model can be used also with datasets that covered areas provided that they contain the same features. We also suggest that people take into consideration the features that were deemed as most important as seen in the previous section; this might help them estimate the customer retention is better.

Learning Outcomes of the Study in respect of Data Science:

- Obtain, clean/process, and transform data.
- Analyze and interpret data using an ethically responsible approach.
- Use appropriate models of analysis, assess the quality of input, derive insight from results, and investigate potential issues.
- Apply computing theory, languages, and algorithms, as well as mathematical and statistical models, and the principles of optimization to appropriately formulate and use data analyses
- Formulate and use appropriate models of data analysis to solve hidden solutions to business-related challenges

Limitations of this work and Scope for Future Work:

There are many things that can be tried to improve the models' predictions. We can create and add more variables, try different models with different subset of features and/or rows, etc. Some of the ideas are listed below:

- Combine the applicants with 1,2,3 or more dependents and make a new feature as discussed in the EDA part.
- Make independent vs independent variable visualizations to discover some more patterns.
- Arrive at the EMI using a better formula which may include interest rates as well.
- Try neural network using TensorFlow or PyTorch.

****