International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 6 Issue 6, September-October 2022 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

A Study on CLTV Model in E-Commerce Domains using Python

Rasamallu Sai Bharath Reddy[1], Dr. T. Narayana Reddy[2]

[1]Student, [2]Assistant Professor,

[1,2]School of Management Studies & JNTU, Anantapur, Andhra Pradesh, India

ABSTRACT

Customer Relationship Management (CRM) system is an information management and analysis tool that can help businesses and other organizations manage their interactions with customers. CRMs were originally designed to target large corporations, but the internet has allowed small business owners to take advantage of these tools as well. Customer data is collected in a CRM database, which allows for advanced analysis such as customer segmentation and contact history. Customer relationship management system (CRMs) is a process in which a business or other organization administers its interactions with customers, typically using data analysis to study large amounts of information. In this article, we will be explaining how you can a E-commerce company can apply their customer relationship management system to analyze their customer base by CLTV, a key marketing metric that allows you to evaluate the impact and outcomes of the firm's customer relationship management strategies and tactics. In order to increase revenue through better marketing campaigns. E-commerce companies consider that customers are their most important asset and that it is essential to estimate the potential value of this asset. Hence, a model for calculating customer's value is essential in these domains. We describe a general modeling approach, based on BG-NBD and Gamma-Gamma models, for calculating customer value in the e-commerce domain. This model extends existing models from the field of direct marketing, by taking into account a sample set of variables required for evaluating customers value in an e-commerce environment. In addition, we present an algorithm for generating this model from historical data, as well as an application of this modeling approach for the creation of a model for e-commerce. This model provides more accurate predictions than existing models regarding the future income generated by customers using Python.

How to cite this paper: Rasamallu Sai Bharath Reddy | Dr. T. Narayana Reddy "A Study on CLTV Model in E-Commerce Domains using Python" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-6 | Issue-6, October 2022, pp.741-759, URL: www.ijtsrd.com/papers/ijtsrd51952.pdf

IJTSRD51952

KEYWORDS: *E-commerce, Customer Life Time Value, Customer Segmentation, Recency, Frequency, Monetary, CLTV model, BG-NBD Model, Data Analytics, Python*

INTRODUCTION

The segmentation of customers according to their customer lifetime value (CLTV) enables companies to adequately build long-term relationships with customers and effectively manage investments into marketing tools. CLTV contributes to solving a number of problems such as decisions related to addressing, retaining and acquiring customers, or issues concerning a company's long-term value (Haenlein et al., 2006). Many different CLTV models were devised in recent decades and, at the same time, the development of ICT gave rise to e-commerce, which is a fast-growing retail market in world. The important part of e-commerce is online shopping, which offers retail sales directly to consumers. Companies engaged in e-commerce have high data availability due to the interactions of customers with their websites and other Internet-based services. The high level of competition, especially in online shopping, drives companies to spend their financial resources on marketing activities as efficiently as possible, which can be helped by implementing a CLTV model that uses available historical data to estimate customer value. However, in their effort to introduce CLTV as a decision-making basis for marketing management, companies operating an online store face the issue of selecting the appropriate CLTV model that would be suitable for their kind of business.

Customers are central to all marketing activities of a company because not only do they generate income, but they increase the company's market value as well. Marketing emphasizes the interconnection of all processes and activities that create, communicate and provide values for customers, including customer relationship management.

In the past two decades, the field of customer relationship management (CRM) went through a significant transformation thanks to information and communication technologies (mainly database and analytical technology). When analyzing customer feedback, companies no longer have to rely only on the aggregated results of quantitative and qualitative research (e.g., questionnaires, focus groups), but they can use their own customer data and concentrate on selected groups or individual customers. This was achieved thanks to the new possibilities of storing and processing available data about individual customers (Jasek et al., 2018).

The CLTV approach forms a bridge between marketing and financial metrics, which means that marketing activities are always related to financial metrics, allowing space for optimization and management (Williams et al., 2015). CLTV shows the way in which (changes in) customer behavior (e.g., increased purchase, retention) can influence future profitability. The relevancy of CLTV applications is leveraged mainly by customer behavior impacting retention, customer-level attributes impacting customer loyalty (e.g., age and gender), and national cultural dimensions affecting the drivers of purchase, frequency and contribution margin. All of these (and other) components used for appropriate CLTV models with available data constitute both direct and indirect influences on CLTV calculations. The main researched applications of CLTV are aimed at the business-to-consumer context while the business-to-business applications are focused on customer asset management(Nenonen et al., 2016).

Customer Relationship Management (CRM):

Customer relationship management (CRM) is a technology for managing all your company's relationships and interactions with customers and potential customers. The goal is simple: Improve business relationships. A CRM system helps companies stay connected to customers, streamline processes, and improve profitability.

When people talk about CRM, they are usually referring to a CRM system, a tool that helps with contact management, sales management, productivity, and more.

A CRM solution helps you focus on your organization's relationships with individual people — including customers, service users, colleagues, or suppliers — throughout your lifecycle with them, including finding new customers, winning their business, and providing support and additional services throughout the relationship.

Customer Lifetime value (CLTV):

Customer Lifetime value (CLTV) is a key marketing metric that allows you to measure the impact and outcomes of the firm's customer relationship management strategies and tactics. CLTV models are used in the field of marketing to evaluate the lifetime value of customers in conventional businesses.

Customer lifetime value (CLTV), is the prediction of a company's net profit contributed to its overall future relationship with a customer. The model can be simple or sophisticated, depending on how complex the predictive analytics techniques are.

Lifetime value is a critical metric because it represents the maximum amount that customers may be expected to spend in order to acquire new ones. As a result, it's crucial in determining the payback of marketing expenses used in marketing mix modeling.

It allows companies to know exactly how much each customer is worth in monetary terms and therefore exactly how much a marketing department should be willing to spend to acquire each customer. It is a concept adopted from direct marketing which looks on the long term customer behavior as the key for success. The norms for this success are based on

- The cost of acquiring a new customer and
- The benefits & costs of retaining an existing customer

Relationship marketing is the key concept that helps the companies in developing loyal consumer base. It embraces all those steps that companies undertake to know and provide value to its customers It is more profitable to have a set of regular long-term loyal and profitable customers than to have more number of customers. The 20:80 rule of marketing is that 20 % of the customer's account for 80% of the company's profits and it is much cheaper to retain a consumer than to attract a new one. To reach these customers and convert them into partners is the real challenge (Ramachandran, *et al.*, 2006).

CLTV Models:

There are several ways (models) you can use to calculate the numbers mentioned above. It's important to understand that different models use and gather different past data a little bit differently. Some are simpler and offer a more general number; others, like the machine learning models, use additional data and a robust algorithm to give you a much more complete overview of your customers and their particularities.

Let's dive into those models a little bit further:

A. Aggregate Model

The Aggregate Model is probably the most common method out there. It has been around the longest, and it is the most straightforward way to calculate CLTV. The aggregate model uses a constant spend rate and churn for all clients. In this method, we have a single CLTV, or in other words, a single group of customers rather than individuals.

Since this model creates one single CLV predicted value, there might be some drawbacks to using it. Because all of your customers are grouped together, you might see a higher customer churn due to seasonality or a higher monetary value for transactions because of a few "big spenders" that influence the overall value.

B. Cohort Model

As the name says, this model uses cohorts to group customers and then calculates the CLTV for each group. The preposition here is based on the principle that customers grouped in a cohort have the same spending patterns or fairly similar behavior patterns.

The algorithm will create cohorts based on your customers' start date (by month). We usually consider the months of the year to group clients since each month usually has different marketing campaigns targeted to reach different kinds of people that, therefore, might be characterized by different behavior patterns.

C. Probabilistic Model

There are several probabilistic models that are commonly used to calculate CLTV predictions. They all use the same data mentioned before (Average Order Value, Purchase Frequency, Customer Churn), but they have slightly different approaches and probability distribution.

Some of the models often used by companies are:

- Pareto/NBD Model
- BG/NBD Model
- MBG/NBD Model
- BG/BB Model
- Gamma-Gamma Model

D. Machine Learning Models

Machine Learning (ML) is an essential Artificial Intelligence (AI) tool that can help predict CLTV with great accuracy. This is because the ML models use algorithms that find patterns in the data you've collected to more accurately forecast future customer behaviors – which has enormous benefits.

As part of the machine learning process, we will also estimate the Recency, Frequency, and Monetary Value (RFM) for the transactions. This helps give you a clear overview of each user's average purchase amount, lifetime duration, and their frequency of purchase.

CLTV models in E-commerce:

For this type of CLTV model to work, you need to have previous data and prepare it so the algorithm can do its job. This process involves removing duplicates and getting rid of empty fields or data that are incorrectly formatted. In the end, it doesn't matter which CLTV model you choose; you will need to prepare and clean up your data first.

The past three decades saw the introduction of a vast number of different models and approaches to calculating CLTV designed for various types of companies, businesses or chosen management views. One of the possible and often mentioned divisions of CLTV models according to the customer-company relationship is into contractual relations (lost for good, retention), semi-contractual relations and non-contractual relations (always a share, migration).

Within the literature were found only two studies in the Web of Science, which include a greater number of comparisons of selected models for the calculation of CLTV based on their empirical research, and therefore a comparison of the predictive capabilities of selected CLTV models on a single dataset on the basis of statistical metrics. (Donkers et al.,2007). analyzed a dataset from an insurance company with contractual settings and concluded that simple profit regression models achieve the best performance (Batislam et al.,2007). used a dataset from a grocery retailer repeatedly focusing on store cards and their usage as the drivers of higher purchase frequency by customers. The results confirm the better performance of their own modified Beta Geometric/NBD model (BG/NBD) customized to the specified business settings in comparison with Pareto/NBD and original BG/NBD models.

It can be stated that even simple models achieve excellent prediction results despite the more complex models being expected to capture the depth of relationship developments better. Similarly, it can be expected that modified models or those designed for specific conditions and environment will produce better predictions in relevant cases than more complex, universally applicable models (achieving consistently good results in various situations). This article focuses on non-contractual relations typical for e-commerce companies engaged in online shopping. Such companies usually have at their disposal an extensive database concerning their customers, which they use for internal purposes (e.g., financial management, marketing). This kind of online retail market, focusing on selling to end customers, has been growing continuously and it can thus be expected that the number of Internet-based services such as online stores will increase. The same applies to the competitive pressure put on them. The focus on e-commerce companies engaged in online shopping is therefore very topical both in local and global context.

NEED OF THE STUDY
- The study is to evaluate CLTV to make better decisions on (CAC) Customer acquisition costs, Improved forecasting, Improving profitability & Strategic marketing practices.
- To make Customer segmentation based on RFM metrics.

OBJECTIVES OF THE STUDY
- To Segment Customers Based on RFM Scores.
- To Segment Customers Based on CLTV.
- To analyse the CLTV in improving Customer Retention and avoiding attrition.
- To Forecast the CLTV to make better decisions on (CAC)Customer acquisition costs.

SCOPE OF THE STUDY
- The study is confined on modeling Customer lifetime value (CLTV) evaluation, marketing metric that projects the value of a customer over the entire history of that customers relationship with a E-Commerce company using BG-NBD statistical models in Python.

LIMITATIONS OF THE STUDY
- Inaccuracy of Data can lead to Misleading results.
- The study is confined on CLTV modeling using BG-NBD statistical models in Python.
- This study is confined to 45days only.

METHODOLOGY OF THE STUDY
This study is entirely based on Secondary Data Analysis, through research papers, journals, articles, websites etc.

Secondary Data Collection:
Sample E-commerce data-set is gathered from a secondary data source through internet.

Source: UCI Machine Learning Repository.

Website: https://archive.ics.uci.edu/ml/index.php

TOOLS AND TECHNIQUES
Software Used:
- Python 3.10
- Jupyter Notebook
- MS Excel

Statistical Models:
BG-NBD and Gamma-Gamma models.

RESEARCH DESIGN
Six Phases Of CLTV Modelling:

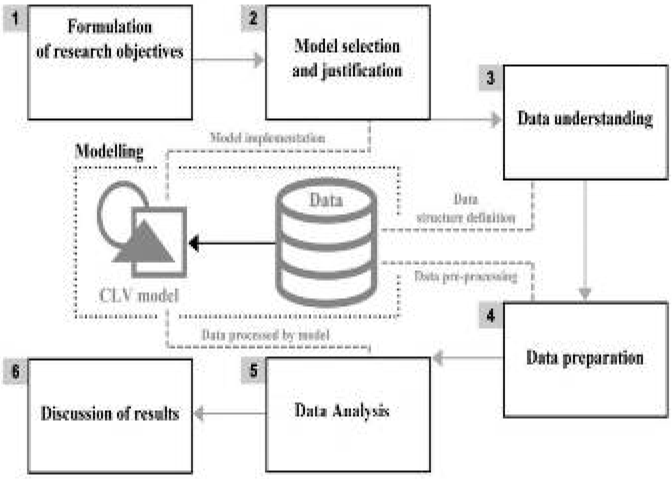

Figure 1: Research Phases CLTV Modelling.

Model Selection And Justification:
According to (Farris et al., 2008), there are several concepts in measuring these values are Customer Profitability Analysis (CPA), Recency and Retention Rate Analysis, and Customer Lifetime Value (CLTV). Researchers recommend CLTV as a metric for selecting customers and designing marketing programs,(Reinartz and Kumar, et al., 2003) and (Rust et al., 2004).

The comparison of CLTV predictive abilities, using selected evaluation metrics, is made on selected CLTV models: Extended Pareto/NBD model (EP/NBD), Markov chain model and Status Quo model. The article uses six online store datasets with annual revenues in the order of tens of millions of euros for the comparison. The EP/NBD model has outperformed other selected models in a majority of evaluation metrics and can be considered good and stable for non-contractual relations in online shopping(Jasek et al., 2018).

DATA COLLECTION ANALYSIS & INTERPRETATION:
DATA UNDERSTANDING:
Dataset:
- The dataset includes Sample sales between 01/12/2009 - 09/12/2011.
- In this article, the years 2010-2011 will be examined.
- The product catalog of this company includes souvenirs.
- The vast majority of the company's customers are corporate customers.

Variables:
- **InvoiceNo:** Invoice number. The unique number of each transaction, namely the invoice. Aborted

operation if it starts with C.

- **StockCode:** Product code. Unique number for each product.
- **Description:** Product name
- **Quantity:** Number of products. It expresses how many of the products on the invoices have been sold.
- **InvoiceDate:** Invoice date and time.
- **UnitPrice:** Product price.
- **CustomerID:** Unique customer number.
- **Country:** The country where the customer lives.

DATA PREPARATION

Steps For Data Preparation:

A. Installing Required Python Libraries.
B. Load and Check Data.
C. Data Pre-processing.
1. Removing Null-Values.
2. Outlier Observations.
3. Exploratory Data Analysis

A. Installing Required Python Libraries:

Python Code For PIP Installing Packages:

- pip install Lifetimes
- pip install openpyxl
- pip install SQLAlchemy
- pip install -U scikit-learn
- pip install squarify
- pip install seaborn
- pip install matplotlib

Python Code For Installing Libraries:

- from sklearn.preprocessing import MinMaxScaler
- from sqlalchemy import create_engine
- from lifetimes import GammaGammaFitter
- from lifetimes import BetaGeoFitter
- from lifetimes. plotting import plot_period_transactions
- import datetime as dt
- import pandas as pd
- import seaborn as sns
- import matplotlib.pyplot as plt
- import squarify
- import warnings
- warnings. filterwarnings("ignore")

B. Load and Check Data:

Python Code For Load Data:

- df=pd.read_excel("c:\\Users\\MyPC\\Downloads\\Online Retail.xlsx")
- df.head()

Output:

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

Table 1: First 5 Instances Of Imported Data-Set.

Python Code For Check Data:

- df = df[~df["Invoice"].str.contains("C", na=False)]
- df.shape
- def check_df(dataframe):
- print(dataframe.shape)
- print(dataframe.columns)
- print(dataframe.dtypes)
- print(dataframe.head())
- print(dataframe.tail())
- print(dataframe.describe().T)
- check_df(df)

C. Data Pre-processing:

1. Removing Null-Values:

Python Code For Removing Null-Values:

- df.isnull().sum()
- df.dropna(inplace=True)
- df.isnull().sum()

2. Outlier Observations:

Python Code For Outlier Observations:

- def outlier_thresholds(dataframe, variable):
- quartile1 = dataframe[variable].quantile(0.01)

- quartile3 = dataframe[variable].quantile(0.99)
- interquantile_range = quartile3 - quartile1
- up_limit = quartile3 + 1.5 * interquantile_range
- low_limit = quartile1 - 1.5 * interquantile_range
- return low_limit, up_limit
- def replace_with_thresholds(dataframe, variable):
- low_limit, up_limit = outlier_thresholds(dataframe, variable)
- dataframe.loc[(dataframe[variable] < low_limit), variable] = low_limit
- dataframe.loc[(dataframe[variable] > up_limit), variable] = up_limit
- replace_with_thresholds(df, "Quantity")
- replace_with_thresholds(df, "Price")

3. Exploratory Data Analysis:

Python Code for Categorical Variables:

- cat_cols = [col for col in df.columns if df[col].dtypes =="O"]
- cat_but_car = [col for col in df.columns if df[col].nunique() > 100 and df[col].dtypes == "O"]
- cat_cols = [col for col in cat_cols if col not in cat_but_car]
- cat_cols

Output: ['Country']

Python Code for Summarizing Categorical Variables:

- def cat_summary(dataframe, col_name, plot=False):
- print(pd.DataFrame({col_name: dataframe[col_name].value_counts(),
- "Ratio": 100 * dataframe[col_name].value_counts() / len(dataframe)}))
- print("##")
- if plot:
- fig_dims = (15, 5)
- fig, ax = plt.subplots(figsize=fig_dims)
- sns.countplot(x=dataframe[col_name], data=dataframe)
- plt.xticks(rotation = 45, ha = 'right')
- plt.show()
- cat_summary(df, "Country", plot=True)

Output:

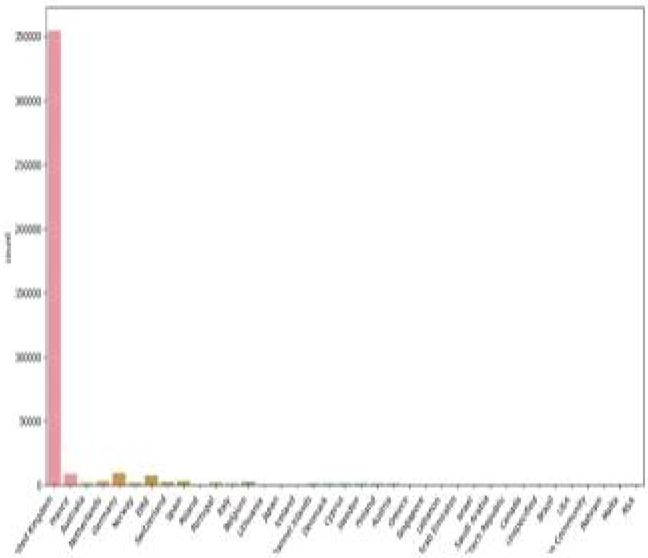

Figure 2: Summarizing Categorical Variable Country.

Interpretation:

The above Bar plot represents the Count of Categorical Variables (Country) where we can observe United kingdom stands first with highest number of transactions, Austria stands last with lowest number of transactions available in Dataset provided.

Python Code For Numerical Variables:

- num_cols = [col for col in df.columns if df[col].dtypes != 'O' and col not in "Customer ID"]
- num_cols

Output:

['Quantity', 'InvoiceDate', 'Price']

Python Code for Summarizing Numerical Variables:

- def num_summary(dataframe, numerical_col, plot=False):
- quantiles = [0.05, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 0.95, 0.99]
- print(dataframe[numerical_col].describe(quantiles).T)
- if plot:
- dataframe[numerical_col].hist(bins=20)
- plt.xlabel(numerical_col)
- plt.title(numerical_col)
- plt.show()
- for col in num_cols:
- num_summary(df, col, plot=True)

Output:

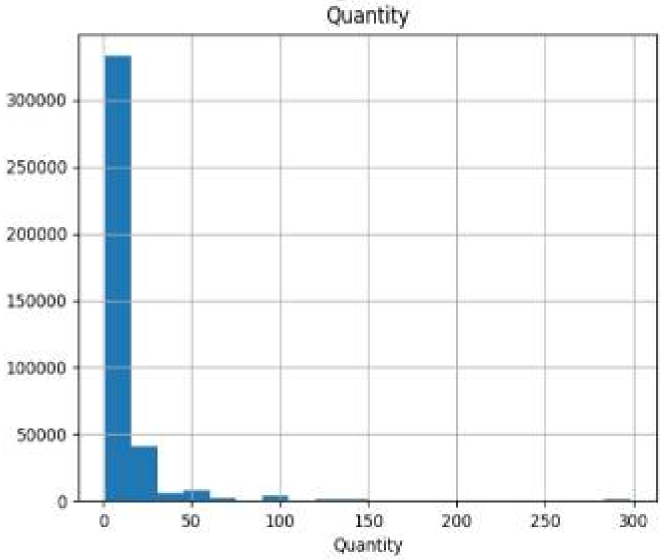

Figure 3: Summarizing Numerical Variable Quantity.

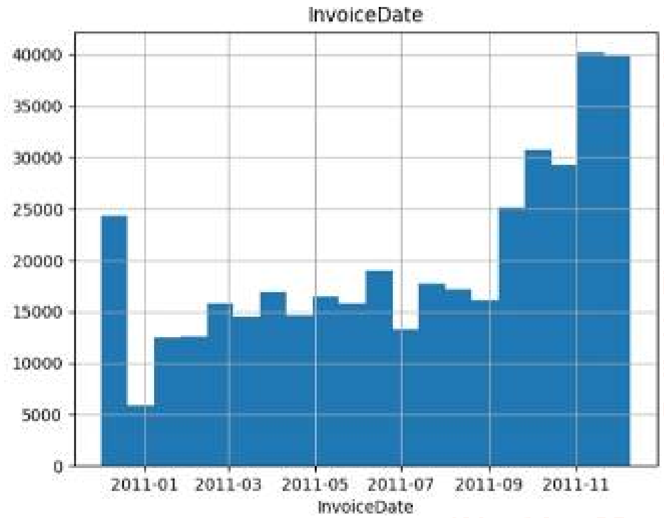

Figure 4: Summarizing Numerical Variable Invoice Date.

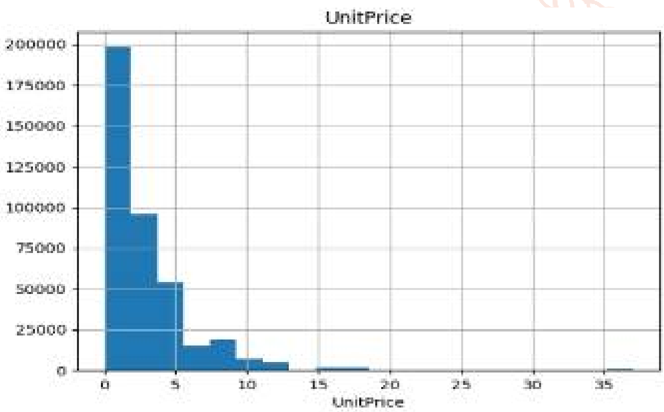

Figure 5: Summarizing Numerical Variable Unit Price.

Interpretation:

The above Bar plots represent the Count of Numerical Variables(Quantity, Invoice date, unit price) where we can observe more than 300000 transactions are made with (0-10) quantity and nearly 10000 transactions are made with (100-150) transactions made available in Dataset provided. More number of transactions are made in Q4. More than 200000 transactions are made with unit price (1-5).

Python Code For How many sales for each product:

➢ df_product = df.groupby("Description").agg({"Quantity":"count"})

➢ df_product.reset_index(inplace=True)

➢ df_product

Output:

	Description	Quantity
0	4 PURPLE FLOCK DINNER CANDLES	39
1	50'S CHRISTMAS GIFT BAG LARGE	109
2	DOLLY GIRL BEAKER	138
3	I LOVE LONDON MINI BACKPACK	70
4	I LOVE LONDON MINI RUCKSACK	1
...	...	...
3872	ZINC T-LIGHT HOLDER STARS SMALL	238
3873	ZINC TOP 2 DOOR WOODEN SHELF	9
3874	ZINC WILLIE WINKIE CANDLE STICK	192
3875	ZINC WIRE KITCHEN ORGANISER	12
3876	ZINC WIRE SWEETHEART LETTER TRAY	20

Table 2: Quantity Sold for each product.

Interpretation:

The above table represents quantity of products sold in serial manner we can observe the product respective quantity available in Dataset provided.

Python Code For Top 10 Products:

➢ top_pr= df_product.sort_values(by="Quantity",ascending=False).head(10)

➢ sns.barplot(x="Description", y="Quantity", data=top_pr)

➢ plt.xticks(rotation=90)

➢ plt.show()

Output:

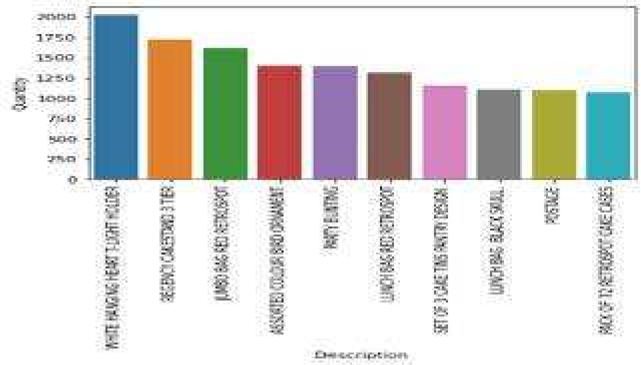

Figure 6: Top 10 Products With Respect To Quantity Sold.

Interpretation:

The above table represents Top 10 Products sorted with respect to Quantity in Dataset provided. We can observe product with description "WHITE

HANGING HEART- LIGHT HOLDER" stands 1st in Top 10 with more than 2000 transactions, "PACK OF 72 RETROSPOT CAKE CASES" stands at 10th.

IMPLEMENTATION OF MODEL FOR DATA ANALYSIS

Steps for Model Implementation:

A. Customer Segmentation With RFM
1. Preparation of RFM Metrics.
2. Generating RFM Scores.
3. Segmenting Customers Based on RFM Scores.
4. Visualization of RFM Segments.

B. Customer Segmentation With CLTV
1. Preparation-Data Structure of CLTV
2. BG-NBD Model
3. Gamma Gamma Model
4. BG-NBD and GG Model For Prediction
5. Segmentation on CLTV Forecasts

A. Customer Segmentation With RFM:

1. Python Code for preparation of RFM Metrics:

total price per invoice

➢ df["TotalPrice"] = df["UnitPrice"] * df["Quantity"]

Determining the analysis date for the recency

➢ df["InvoiceDate"] = pd.to_datetime(df["InvoiceDate"])

➢ df["InvoiceDate"].max()

➢ today_date = dt.datetime(2011, 12, 11)

Generating RFM metrics

➢ rfm = df.groupby("CustomerID").agg({"InvoiceDate": lambda Date: (today_date- Date.max()).days,

"InvoiceNo": lambda Invoice: Invoice.nunique(),

"TotalPrice": lambda TotalPrice: TotalPrice.sum()})

➢ rfm.columns = ["recency","frequency","monetary"]

➢ rfm.head()

➢ rfm.describe().T

Output:

	recency	frequency	monetary
CustomerID			
12346.0	326	1	310.44
12347.0	3	7	4310.00
12348.0	76	4	1770.78
12349.0	19	1	1491.72
12350.0	311	1	331.46

Table 3: RFM Metrics Generated for Each Customer ID.

	count	mean	std	min	25%	50%	75%	max
recency	4339.0	93.041484	100.0077 57	1.0	18.000	51.00	142.500	374.000
frequency	4339.0	4.271952	7.705493	1.0	1.000	2.00	5.000	210.000
monetary	4339.0	1891.7439 68	7705.372 627	0.0	303.125	662.59	1630.445	266163.525

Table 4: Statistical Description of RFM Metrics.

Interpretation:

The above table represents top 5 instances of generated RFM(Recency, Frequency, Monetary) metrics in Dataset provided. And also Describes the RFM metrics(count, mean, std, min, max).

Where we can observe "min of Monetary" is 0.0 & "max of Monetary" is 266163.525

monetary, the min value of the total money paid can't be 0.

Code:

let's remove them from the data

➢ rfm = rfm[rfm["monetary"] > 0]

➢ rfm.describe().T

Output:

	count	mean	std	min	25%	50%	75%	max
recency	4338.0	93.059474	100.012264	1.00	18.0000	51.0	142.7500	374.000
frequency	4338.0	4.272706	7.706221	1.00	1.0000	2.0	5.0000	210.000
monetary	4338.0	1892.184204	7706.206805	3.75	303.3075	663.1	1631.1075	266163.525

Table 5: Statistical Description of Corrected RFM Metrics.

Interpretation:

The above table represents "minimum of monetary value 0" is replaced with "minimum value greater than 0" 3.75 in Dataset provided.

- **Python Code for Generating RFM Scores**

recency_score

- rfm["recency_score"] = pd.qcut(rfm['recency'], 5, labels=[5, 4, 3, 2, 1])

frequency_score

- rfm["frequency_score"]=pd.qcut(rfm["frequency"].rank(method="first"), 5, labels=[1, 2, 3, 4, 5])

monetary_score

- rfm["monetary_score"] = pd.qcut(rfm["monetary"], 5, labels=[1, 2, 3, 4, 5])

RFM Score

- rfm["RFM_SCORE"]=(rfm["recency_score"].astype(str)+ rfm["frequency_score"].astype(str))
- rfm.head(10)

Output:

CustomerID	recency	frequency	monetary	recency_score	frequency_score	monetary_score	RFM_SCORE
12346.0	326	1	310.44	1	1	2	11
12347.0	3	7	4310.00	5	5	5	55
12348.0	76	4	1770.78	2	4	4	24
12349.0	19	1	1491.72	4	1	4	41
12350.0	311	1	331.46	1	1	2	11
12352.0	37	8	1756.34	3	5	4	35
12353.0	205	1	89.00	1	1	1	11
12354.0	233	1	1079.40	1	1	4	11
12355.0	215	1	459.40	1	1	2	11
12356.0	23	3	2811.43	4	3	5	43

Table 6: Generated RFM Scores for Each Customer ID.

Interpretation:

The above table represents 10 instances of generated RFM scores in Dataset provided. We can observe the RFM scores of each "Customer ID" individualy.

- **Python Code for Segmenting Customers Based on RFM Scores**
- seg_map = {
- r'[1-2][1-2]': 'hibernating',
- r'[1-2][3-4]': 'at_Risk',
- r'[1-2]5': 'cant_loose',
- r'3[1-2]': 'about_to_sleep',

- r'33': 'need_attention',
- r'[3-4][4-5]': 'loyal_customers',
- r'41': 'promising',
- r'51': 'new_customers',
- r'[4-5][2-3]': 'potential_loyalists',
- r'5[4-5]': 'champions'
- }
- rfm['segment'] = rfm['RFM_SCORE'].replace(seg_map, regex=True)
- rfm.head(10)

Output:

ustomerID	recency	frequency	monetary	recency_score	frequency_score	monetary_score	RFM_SCORE	segment
12346.0	326	1	310.44	1	1	2	11	hibernating
12347.0	3	7	4310.00	5	5	5	55	champions
12348.0	76	4	1770.78	2	4	4	24	at_Risk
12349.0	19	1	1491.72	4	1	4	41	promising
12350.0	311	1	331.46	1	1	2	11	hibernating
12352.0	37	8	1756.34	3	5	4	35	loyal_customers
12353.0	205	1	89.00	1	1	1	11	hibernating
12354.0	233	1	1079.40	1	1	4	11	hibernating
12355.0	215	1	459.40	1	1	2	11	hibernating
12356.0	23	3	2811.43	4	3	5	43	potential_lo

Table 7: Customers Segmentation based on generated RFM scores.

Interpretation:
The above table represents Top 10 instances of Customers Segmented based on generated RFM scores in Dataset provided. We can observe the Customers Segmented based on generated RFM scores of each "Customer ID" individualy.

Python Code for grouping RFM mean and count values according to segments

- rfm[["segment","recency","frequency","monetary"]].groupby("segment").agg(["mean", "count"])

Output:

segment	recency		frequency		monetary	
	mean	count	mean	count	mean	count
about_to_sleep	53.312500	352	1.161932	352	469.058097	352
at_Risk	153.785835	593	2.878583	593	938.456341	593
cant_loose	132.968254	63	8.380952	63	2646.822540	63
champions	6.361769	633	12.417062	633	6498.612978	633
hibernating	217.605042	1071	1.101774	1071	398.573036	1071
loyal_customers	33.608059	819	6.479853	819	2752.519574	819
need_attention	52.427807	187	2.326203	187	847.657086	187
new_customers	7.428571	42	1.000000	42	314.883690	42
potential_loyalists	17.398760	484	2.010331	484	674.628357	484
promising	23.510638	94	1.000000	96	285.623723	94

Table 8: RFM mean and count values according to Customer Segments

Interpretation:
The above table represents Count, Mean of RFM Segments generated RFM scores in Dataset provided. We can observe number of Hibernating customers is "1071", Loyal_customers is "819", Champions customers is "633", At_Risk customers is "593", Potential_loyalists customers is "484", About_to_sleep customers is "352", Need_attention customers is "187", Promising customers is "94", Cant_loose customers is "63".

➢ **Python code for Visualization of RFM Segments**

- sgm= rfm["segment"].value_counts()
- plt.figure(figsize=(10,7))
- sns.barplot(x=sgm.index,y=sgm.values)
- plt.xticks(rotation=45)
- plt.title('Customer Segments',color = 'blue',fontsize=15)
- plt.show()

Output:

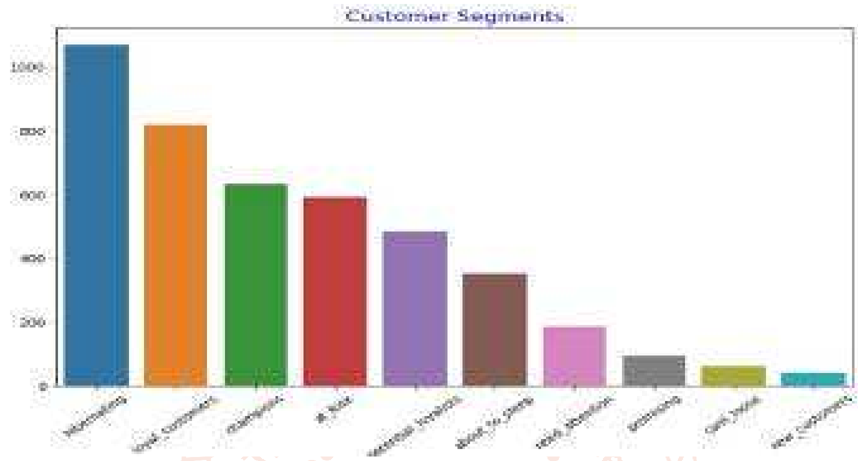

Figure 7: Bar Plot Of Customer Segmentation Based On RFM Scores.

Interpretation:

The above bar plot represents Visualization of Customer Segmentation based on RFM scores generated in Dataset provided. We can observe graph showing descending bar plot of Hibernating customers is “1”, Loyal_customers is “2”, Champions customers is “3”, At_Risk customers is “4”, Potential_loyalists customers is “5”, About_to_sleep customers is “6”, Need_attention customers is “7”, Promising customers is “8”, Cant_loose customers is “9”, New_customers is “10”.

Python Code forTreemap Visualization

➢ df_treemap = rfm.groupby('segment').agg('count').reset_index()

➢ df_treemap.head()

Output:

	segment	recency	frequency	monetary	recency_score	frequency_score	monetary_score	RFM_SCORE
0	about_to_sleep	352	352	352	352	352	352	352
1	at_Risk	593	593	593	593	593	593	593
2	cant_loose	63	63	63	63	63	63	63
3	champions	633	633	633	633	633	633	633
4	hibernating	1071	1071	1071	1071	1071	1071	1071

Table 9: Customer Segmentation Count Based On RFM Scores.

Interpretation:

The above table represents count of Customer Segmentation based on RFM scores generated in Dataset provided. We can observe that number of about_to_sleep is 352, at_Risk is 593, cant_loose is 63, champions is 633, hibernating is 1071.

Python Code forPlotting Treemap

➢ fig, ax = plt.subplots(1, figsize = (16,10))

➢ squarify.plot(sizes=df_treemap['RFM_SCORE'], label=df_treemap['segment'],

alpha=.8,

color=['tab:red', 'tab:purple', 'tab:brown', 'tab:pink', 'tab:gray']

)

- plt.axis('off')
- plt.show()
- plt.savefig('treemap.png')

Output:

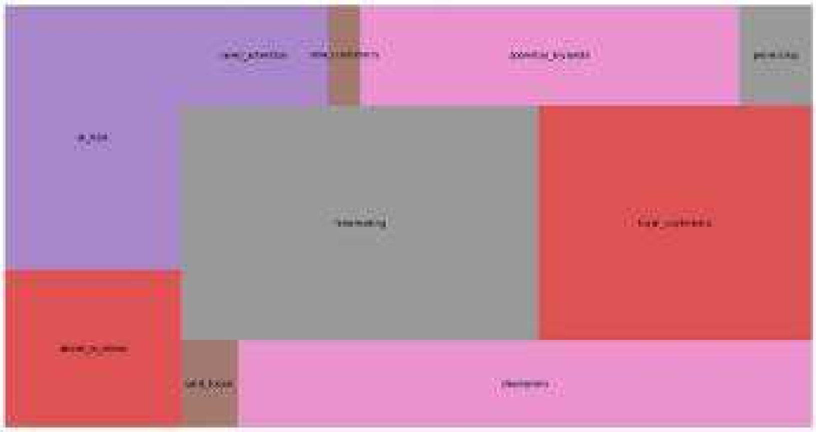

Figure 8: Tree Map Visualization Of Customer Segmentation.

Interpretation:

The above visualization represents tree map of Customer Segmentation based on RFM scores generated in Dataset provided. We can observe the Customer segmentation in a glance by using Tree Map.

A. Customer Segmentation With CLTV:

1. Python Code for Preparation-Data Structure of CLTV

Determining the analysis date for the recency

- df["InvoiceDate"] = pd.to_datetime(df["InvoiceDate"])
- df["InvoiceDate"].max()
- today_date = dt.datetime(2011, 12, 11)

Generating CLTV metrics

- cltv_df = df.groupby('CustomerID').agg({'InvoiceDate': [lambda date: (date.max() - date.min()).days,

lambdadate:(today_date-date.min()).days],

'InvoiceNo': lambda num: num.nunique(),

'TotalPrice': lambda TotalPrice: TotalPrice.sum()})

- cltv_df.columns = cltv_df.columns.droplevel(0)
- cltv_df.columns = ['recency', 'T', 'frequency', 'monetary']
- cltv_df.head()

Output:

CustomerID	recency	T	frequency	monetary
12346.0	0	326	1	310.44
12347.0	365	368	7	4310.00
12348.0	282	359	4	1770.78
12349.0	0	19	1	1491.72
12350.0	0	311	1	331.46

Table 10: RFTM Values of Each Customer ID.

Code:

we calculated the monetary values as the Total price.

At this point we will express the monetary value as the average earnings per purchase.

➢ cltv_df["monetary"] = cltv_df["monetary"] / cltv_df["frequency"]

selection of monetary values greater that Zero.

➢ cltv_df = cltv_df[cltv_df["monetary"] > 0]

Weekly expression of Recency and T for BGNBD

➢ cltv_df["recency"] = cltv_df["recency"] / 7

➢ cltv_df["T"] = cltv_df["T"] / 7

Selecting Frequency greater than 1

➢ cltv_df = cltv_df[(cltv_df['frequency'] > 1)]

➢ cltv_df.head()

Output:

CustomerID	recency	T	frequency	monetary
12347.0	52.142857	52.571429	7	615.714286
12348.0	40.285714	51.285714	4	442.695000
12352.0	37.142857	42.428571	8	219.542500
12356.0	43.142857	46.571429	3	937.143333
12358.0	21.285714	21.571429	2	575.210000

Table 11: Updated RFTM Values of Each Customer ID.

Interpretation:

The above table represents data structure prepared for generating CLTV based on RFM Metrices. Here we have added a new columns "T" which represents "Tenure" of every individual cutomer ID.

➢ **Python Code for BG-NBD Model**

For modelling, we first fit our Frequency, Recency and T columns to BG/NBD model.

➢ bgf = BetaGeoFitter(penalizer_coef=0.001)

➢ bgf.fit(cltv_df['frequency'],

➢ cltv_df['recency'],

➢ cltv_df['T'])

1 week expected purchase (transaction)

➢ cltv_df["expected_purc_1_week"] = bgf.predict(1,

cltv_df['frequency'],

cltv_df['recency'],

cltv_df['T'])

➢ cltv_df.sort_values("expected_purc_1_week", ascending=False).head(10)

Output:

CustomerID	recency	T	frequency	monetary	expected_purc_1_week
12748.0	53.142857	53.428571	210	154.192429	3.265158
14911.0	53.142857	53.428571	201	691.710100	3.126645
17841.0	53.000000	53.428571	124	330.134355	1.940290
13089.0	52.285714	52.857143	97	606.362474	1.537528
14606.0	53.142857	53.428571	93	130.139032	1.463999
15311.0	53.285714	53.428571	91	667.779121	1.433717
12971.0	52.571429	53.285714	86	127.485872	1.357024
14646.0	50.428571	50.714286	74	3596.804392	1.222517
13408.0	53.000000	53.428571	62	453.500645	0.986249
18102.0	52.285714	52.571429	60	3859.739083	0.968607

Table 12 : Expected Purchase Rate For 1 Week, Generated by Fitting BG-NBD Model based on RFM Metrices.

Interpretation:
The above table represents expected purchase rate for 1 week, generated by fitting BG-NBD Model based on RFM Metrices. We can observe the expected purchase for 1 week of every individual with respective "Customer ID".

Python code for 1 month expected purchase
- ➢ cltv_df["expected_purc_1_month"] = bgf.predict(4, cltv_df['frequency'], cltv_df['recency'], cltv_df['T'])
- ➢ cltv_df.sort_values("expected_purc_1_month", ascending=False).head(10)

Output:

CustomerID	recency	T	frequency	monetary	expected_purc_1_week	expected_purc_1_month
12748.0	53.142857	53.428571	210	154.192429	3.265158	13.025670
14911.0	53.142857	53.428571	201	691.710100	3.126645	12.473095
17841.0	53.000000	53.428571	124	330.134355	1.940290	7.740945
13089.0	52.285714	52.857143	97	606.362474	1.537528	6.133456
14606.0	53.142857	53.428571	93	130.139032	1.463999	5.840269
15311.0	53.285714	53.428571	91	667.779121	1.433717	5.719467
12971.0	52.571429	53.285714	86	127.485872	1.357024	5.413481
14646.0	50.428571	50.714286	74	3596.804392	1.222517	4.876360
13408.0	53.000000	53.428571	62	453.500645	0.986249	3.934373
18102.0	52.285714	52.571429	60	3859.739083	0.968607	3.863856

Table 13 : Expected Purchase Rate For 1 Month, Generated by Fitting BG-NBD Model based on RFM Metrices.

Interpretation:
The above table represents expected purchase for 1 month, generated by fitting BG-NBD Model based on RFM Metrices.

➢ **Python Code for Gamma Gamma Model**

For modelling, we first fit our Frequency, Recency columns to Gamma Gamma Submodel

➢ ggf = GammaGammaFitter (penalizer_coef=0.01)

➢ ggf.fit(cltv_df['frequency'], cltv_df['monetary'])

Expected average profit

➢ cltv_df["expected_average_profit"]=ggf.conditional_expected_average_profit

(cltv_df['frequency'], cltv_df['monetary'])

➢ cltv_df.sort_values("expected_average_profit", ascending=False).head(20)

Output:

CustomerID	recency	T	frequency	monetary	expected_purc_1_week	expected_purc_1_month	expected_average_profit
12415.0	44.714286	48.285714	21	5724.302619	0.379618	1.513975	5772.177098
12590.0	0.000000	30.285714	2	4591.173500	0.011536	0.045934	5029.408953
12435.0	26.857143	38.285714	2	3914.945000	0.076313	0.304072	4288.934027
12409.0	14.714286	26.142857	3	3690.890000	0.117435	0.467806	3918.807445
14088.0	44.571429	46.142857	13	3864.554615	0.260297	1.037956	3917.128538
18102.0	52.285714	52.571429	60	3859.739083	0.968607	3.863856	3870.996680
12753.0	48.428571	51.857143	6	3571.565000	0.126058	0.502765	3678.579903
14646.0	50.428571	50.714286	74	3596.804392	1.222517	4.876360	3605.309143
15749.0	13.857143	47.571429	3	3028.780000	0.027979	0.111556	3219.047938
14096.0	13.857143	14.571429	17	3163.588235	0.728797	2.895990	3196.435322
17511.0	52.857143	53.428571	31	2933.943065	0.508850	2.029876	2950.579762
17450.0	51.285714	52.571429	46	2863.274891	0.747475	2.981719	2874.198442
13081.0	51.285714	53.142857	11	2576.125455	0.201222	0.802640	2617.760374
16000.0	0.000000	0.428571	3	2335.120000	0.422126	1.664397	2473.801453
16684.0	5.857143	18.714286	2	2240.675000	0.102765	0.408817	2455.597113

Table 14 : Expected Average Profit, Generated by Fitting BG-NBD Model based on RFM Metrices.

Interpretation:

The above table represents expected average profit, which in other words Customer Acquistion Cost generated by fitting Gamma Gamma Submodel based on RFM Metrices.

- **Python Code for BG-NBD and GG Model For Prediction**

- cltv = ggf.customer_lifetime_value(bgf,
- cltv_df['frequency'],
- cltv_df['recency'],
- cltv_df['T'],
- cltv_df['monetary'],
- time=6, # 6 months.
- freq="W", # T's Frequency information.
- discount_rate=0.01)

Reset index

- cltv = cltv.reset_index()

Merging the main table and the forecast values table

- cltv_final = cltv_df.merge(cltv, on="Customer ID", how="left")

sorting

- cltv_final.sort_values(by="clv", ascending=False).head(10)

Output:

	Customer ID	recency	T	frequency	monetary	expected_purc_1_week	expected_purc_1_month	expected_average_profit	index	clv
1122	14646.0	50.428571	50.714286	74	3596.804392	1.222517	4.876360	3605.309143	1122	108691.797610
2761	18102.0	52.285714	52.571429	60	3859.739083	0.968607	3.863856	3870.996680	2761	92510.277355
843	14096.0	13.857143	14.571429	17	3163.588235	0.728797	2.895990	3196.435322	843	56135.699371
36	12415.0	44.714286	48.285714	21	5724.302619	0.379618	1.513975	5772.177098	36	53978.796679
1257	14911.0	53.142857	53.428571	201	691.710100	3.126645	12.473095	692.326854	1257	53427.082011
2458	17450.0	51.285714	52.571429	46	2863.274891	0.747475	2.981719	2874.198442	2458	53004.426445
874	14156.0	51.571429	53.142857	55	2104.021077	0.877551	3.500701	2110.754078	874	45708.064212
2487	17511.0	52.857143	53.428571	31	2933.943065	0.508850	2.029876	2950.579762	2487	37047.559984
2075	16684.0	50.428571	51.285714	28	2209.969107	0.475081	1.926950	2223.884664	2075	26217.822999
650	13694.0	52.714286	53.428571	50	1275.700500	0.800854	3.194773	1280.218159	650	25301.558007

Table 15 : CLTV Generated by Fitting BG-NBD Model based on RFM Metrices.

Interpretation:

- The above table represents CLV, generated by BG-NBD and GG Model Prediction based on RFM Metrices.

Python Code for 12 Month CLTV Prediction:

12 Month CLTV Forecast:

- cltv_12 = ggf.customer_lifetime_value(bgf,
- cltv_df['frequency'],
- cltv_df['recency'],
- cltv_df['T'],
- cltv_df['monetary'],
- time=12, # 1 aylık
- freq="W", # T'nin frekans bilgisi
- discount_rate=0.01)
- cltv_12.head()
- cltv_12 = cltv_12.reset_index()
- cltv_12 = cltv_df.merge(cltv_12, on="CustomerID", how="left")
- cltv_12.sort_values(by="clv", ascending=False).head(10)

Output:

	CustomerID	recency	T	frequency	monetary	expected_purc_1_week	expected_purc_1_month	expected_average_profit	clv
1122	14646.0	50.428571	50.714286	74	3596.804392	1.222517	4.876360	3605.309143	207476.974690
2761	18102.0	52.285714	52.571429	60	3859.739083	0.968607	3.863856	3870.996680	178649.341155
843	14096.0	13.857143	14.571429	17	3163.588235	0.728797	2.895990	3196.435322	105762.288258
36	12415.0	44.714286	48.285714	21	5724.302619	0.379618	1.513975	5772.177098	102964.325323
1257	14911.0	53.142857	53.428571	201	691.710100	3.126645	12.473095	692.326354	102046.017793
2458	17450.0	51.285714	52.571429	46	2863.274891	0.747475	2.981729	2874.198442	101206.239080
874	14156.0	51.571429	53.142857	55	2104.026727	0.877551	3.500701	2110.754078	87288.674040
2487	17511.0	52.857143	53.428571	31	2933.943065	0.508850	2.029876	2950.579762	70747.401801
2075	16684.0	50.428571	51.285714	28	2209.969107	0.478081	1.905990	2223.884664	50094.759622
650	13694.0	52.714286	53.428571	50	1275.700500	0.820854	3.194773	1282.218159	48320.424234

Table 16 : 12 Month CLTV Forecast, Generated by Fitting BG-NBD Model based on RFM Metrices.

Interpretation:

The above table represents 12 Month CLTV Prediction, generated by BG-NBD and GG Model Prediction based on RFM Metrices.

- **Python Code for Segmentation on CLTV Forecasts**

Python code for Normalization 0-1 Range For CLV Values

- scaler = MinMaxScaler(feature_range=(0, 1))
- scaler.fit(cltv_final[["clv"]])
- cltv_final["scaled_clv"] = scaler.transform(cltv_final[["clv"]])
- cltv_final.sort_values(by="scaled_clv", ascending=False).head()

Output:

	CustomerID	recency	T	frequency	monetary	expected_purc_1_week	expected_purc_1_month	expected_average_profit	index	clv	scaled_clv
1122	14646.0	50.428571	50.714286	74	3596.804392	1.222517	4.876360	3605.309143	1122	108691.757610	1.000000
2761	18102.0	52.285714	52.571429	60	3859.739083	0.968607	3.863856	3870.996680	2761	92510.277355	0.851125
843	14096.0	13.857143	14.571429	17	3163.588235	0.728797	2.895990	3196.435322	843	56135.699371	0.516467
36	12415.0	44.714286	48.285714	21	5724.302619	0.379618	1.513975	5772.177098	36	53978.798679	0.496623
1257	14911.0	53.142857	53.428571	201	691.710100	3.126645	12.473095	692.326354	1257	53427.083011	0.491547

Table 17 : Scaled CLTV Rate, Generated by Fitting BG-NBD Model based on RFM Metrices.

Interpretation:
The above table represents scaled CLV values generated based on RFM metrices in Dataset provided.

Python code for Segmentation of Customers

- cltv_final["segment"] = pd.qcut(cltv_final["scaled_clv"], 4, labels=["D", "C", "B", "A"])
- cltv_final.head()
- cltv_final.head()

Output:

	CustomerID	recency	T	frequency	monetary	expected_purc_1_week	expected_purc_1_month	expected_average_profit	index	clv	scaled_clv	segment
0	12347.0	52.142857	52.571429	7	615.714286	0.141283	0.553516	631.911943	0	2200.757832	0.020248	A
1	12348.0	40.285714	51.285714	4	442.695000	0.091963	0.366752	463.745603	1	1050.352232	0.009664	B
2	12352.0	37.142857	42.428571	8	219.542500	0.182395	0.727126	224.886660	2	1007.740765	0.009272	B
3	12356.0	43.142857	46.571429	3	937.143333	0.086160	0.343511	995.997563	3	2109.557658	0.019409	A
4	12358.0	21.285714	21.571429	2	575.210000	0.122258	0.486232	631.900837	4	1870.554771	0.017210	A

Table 18 : Customer Segmentation based on CLTV Forecasted Values.

Interpretation:
The above table represents labelling segments of based on scaled CLV generated in Dataset provided.

Python code for Examination of Segments by count

- cltv_final.groupby("segment").agg({"count"})

Output:

segment	Customer ID	recency	T	frequency	monetary	expected_purc_1_week	expected_purc_1_month	expected_average_profit	clv	scaled_clv
	count	count	count	count	count	count	count	count	count	count
D	712	712	712	712	712	712	712	712	712	712
C	711	711	711	711	711	711	711	711	711	711
B	711	711	711	711	711	711	711	711	711	711
A	711	711	711	711	711	711	711	711	711	711

Table 19 : Count of Customer Segments based on CLTV Forecasted Values.

Interpretation:
The above table represents the count of Segment labels based on CLV generated in Dataset provided.

Python code for Examination of Segments by sum

- cltv_final.groupby("segment").agg({"sum"})

Output:

segment	Customer ID	recency	T	frequency	monetary	expected_purc_1_week	expected_purc_1_month	expected_average_profit	clv	scaled_clv
	sum	sum	sum	sum	sum	sum	sum	sum	sum	sum
D	11073468.0	15705.142857	28760.285714	2183	110733.013924	50.692720	201.885835	141735.766747	1.983117e+05	1.824532
C	10895576.0	21747.142857	27054.428571	2509	193055.954775	85.878469	341.975364	206064.368539	5.261275e+05	4.840544
B	10931050.0	20974.142857	24746.428571	3856	265857.464706	115.264861	458.727525	280451.771609	9.498403e+05	8.738840
A	10626242.0	22414.428571	24556.142857	8094	469090.056231	194.554052	774.556345	487558.459248	3.069753e+06	28.242727

Table 20 : Sum of Customer Segments based on CLTV Forecasted Values.

Interpretation:
The above table represents the sum of Segment labels based on CLV generated in Data-set provided.

Python code for Examination of Segments by mean

➢ cltv_final.groupby("segment").agg({"mean"})

Output:

segment	Customer ID	recency	T	frequency	monetary	expected_purc_1_week	expected_purc_1_month	expected_average_profit	clv	scaled_clv
	mean	mean	mean	mean	mean	mean	mean	mean	mean	mean
D	15552.651685	22.057785	40.421750	3.066011	183.613784	0.071198	0.283548	199.067088	278.527600	0.002563
C	15324.298172	30.586699	38.051236	4.091421	271.527363	0.120785	0.480978	289.823303	739.932380	0.006808
B	15345.857947	29.459498	34.805103	5.423347	373.920485	0.162117	0.645186	394.446936	1335.321627	0.012291
A	14945.488045	31.525216	34.593731	11.383966	659.761035	0.273634	1.089390	685.736230	4317.514384	0.039723

Table 21 : Mean of Customer Segments based on CLTV Forecasted Values.

Interpretation:
The above table represents the sum of Segment labels based on CLV generated in Dataset provided.

FINDINGS OF THE STUDY

- ➢ Most number of the transactions are made from United Kingdom, Least number of transactions are made from Austria.
- ➢ Highest number of transactions are made between dates (2011/09 to 2011/11).
- ➢ Top 10 products being bought by the Customers.
- ➢ Based on RFM metrics Number of Hibernating customers is "1071".
- ➢ Loyal_customers count based on RFM metrics is "819".
- ➢ Champions customers count based on RFM metrics is "633".
- ➢ At_Risk customers count based on RFM metrics is "593".
- ➢ Potential_loyalists customers count based on RFM metrics is "484".
- ➢ About_to_sleep customers count based on RFM metrics is "352".
- ➢ Need_attention customers count based on RFM metrics is "187".
- ➢ Promising customers count based on RFM metrices is "94".
- ➢ Cant_loose customers count based on RFM metrices is "63".
- ➢ New_customers count based on RFM metrics is "42".
- ➢ Based on CLTV metrics Segments "B" group customers generate most CLTV than any other segments.
- ➢ Based on CLTV metrics Segments "D" group customers generate least CLTV than any other segments.
- ➢ Based on CLTV metrics Segments "A" group customers generate more Frequency than any other segments.
- ➢ Based on CLTV metrics Segments "A" group customers generate more Expected Average Profit than any other segments.

SUGGESTIONS

- ➢ Running Customer based campaigns for "C" segment customers can increase more CLV and Customer Acquisition Cost (CAC) can be reduced when compared with "D" segment customers.
- ➢ Running Loyalty based campaigns for "A" segment customers can help generate more CLV and reduce Customer Acquisition Cost (CAC) when compared with "B" segment customers.
- ➢ Concentrating on customer who are Hibernating customers, About_to_sleep customers can help in improving "Customer Retention".
- ➢ Concentrating on customer who are About_to_sleep customers, Need_attention customers can help in reducing "Customer Attrition".

CONCLUSION

- ➢ CLTV model helps you determine how much money you can afford to spend acquiring new customers and retaining existing ones and RFM can be used to segment your customers to better target your marketing efforts. All the efforts to perform these analyses have only one goal which is to make better business decisions based on data. Using CLTV and RFM simultaneously and interpreting data based on both analyses can help businesses grow.

- Performing these analyses using Python and Lifetimes modules is not the most important and complicated part. Knowing your business, having domain knowledge and using that knowledge to produce useful results from these analyses is the key to a business's success.
- It's not wise to serve all customers with the same product model, email, text message campaign, or ad. Customers have different needs. A one-size-for-all approach to business will generally result in less engagement, lower-click through rates, and ultimately fewer sales. Customer segmentation is the cure for this problem.
- Finding an optimal number of unique customer groups will help you understand how your customers differ, and help you give them exactly what they want. Customer segmentation improves customer experience and boosts company revenue. That's why segmentation is a must if you want to surpass your competitors and get more customers. Doing it with machine learning is definitely the right way to go.

BIBLIOGRAPHY

References:

[1] Jašek, Pavel & Vraná, Lenka & Sperkova, Lucie & Smutny, Zdenek & Kobulsky, Marek. (2018). Modeling and Application of Customer Lifetime Value in Online Retail. Informatics. 5. 10.3390/informatics5010002.

[2] Haenlein, M.; Kaplan, A.M.; Schoder, D. Valuing the Real Option of Abandoning Unprofitable Customers When Calculating Customer Lifetime Value. J. Mark. 2006, 70, 5–20.

[3] Kumar, V.; Pozza, I.D.; Petersen, J.A.; Denish Shah, D. Reversing the Logic: The Path to Profitability through Relationship Marketing. J. Int. Mark. 2009, 23, 147–156.

[4] Doyle, P. Value-based marketing. J. Strateg. Mark. 2000, 8, 299–311

[5] Srivastava, R.K.; Shervani, T.A.; Fahey, L. Market-Based Assets and Shareholder Value: A Framework for Analysis. J. Mark. 1998, 62, 2–18.

[6] Gupta, S. Customer-Based Valuation. J. Interact. Mark. 2009, 23, 169–178.

[7] Persson, A.; Ryals, L. Customer assets and customer equity: Management and measurement issues. Mark. Theory 2010, 10, 417–436.

[8] Ryals, L.; Knox, S. Measuring and managing customer relationship risk in business markets. Ind. Mark. Manag. 2007, 36, 823–833.

[9] Williams, C.; Williams, R. Optimizing acquisition and retention spending to maximize market share. J. Mark. Anal. 2015, 3, 159–170.

[10] Nenonen, S. Storbacka, K. Driving shareholder value with customer asset management: Moving beyond customer lifetime value. Ind. Mark. Manag. 2016, 52, 140–150

[11] Donkers, B.; Verhoef, P.C.; De Jong, M.G. Modeling CLV: A test of competing models in the insurance industry. Quant. Mark. Econ. 2007, 5, 163–190.

[12] Batislam, E.M.; Denizel, M.; Filiztekin, A. Empirical Validation and Comparison of Models for Customer Base Analysis. Int. J. Res. Mark. 2007, 24, 201–209.

[13] Farris P. W., Bendle N. T., Pfeifer P. E., and Reibstein D. J. 2008. Marketing Metric. Jakarta: Akademia.

[14] Rust, Roland T., Valarie A. Zeithaml, and Katherine N. Lemon. 2004. Return on Marketing: Using Customer Equity to Focus Marketing Strategy. Journal of Marketing, 68 (January), 23–53.

[15] Ramachandran, Ramakrishnan. (2006). Customer Lifetime Value. 10.13140/2.1.1787.1049.

Books:

[1] Kotler, P.; Keller, K.L. Marketing Management, 15th ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2015.

[2] Fader, P.S. Customer Centricity: Focus on the Right Customers for Strategic Advantage; Wharton Digital Press: Philadelphia, PA, USA, 2012.

Web:

[1] Centre for Retail Research. Available online: http://www.retailresearch.org/onlineretailing.php

[2] Indian Brand Equity Foundation: https://www.ibef.org/industry/ecommerce