Determining Your Most Valuable Customers

CUSTOMER ANALYSIS

1155192555 Choi Chit
1155191541 Jiang Xinwen
1155193869 Tang Yu Ching
1155191528 Zhang Yuchen
1155193163 Chong Chun Hin
1155193808 Ng Yuk
1155191554 Wu Tong

Group 1

CUSTOMER ANALYSIS

A Introduction

B Original solution

C Extension 1
-- Gamma Gamma

D Extension 2
-- K-means

E Conclusion

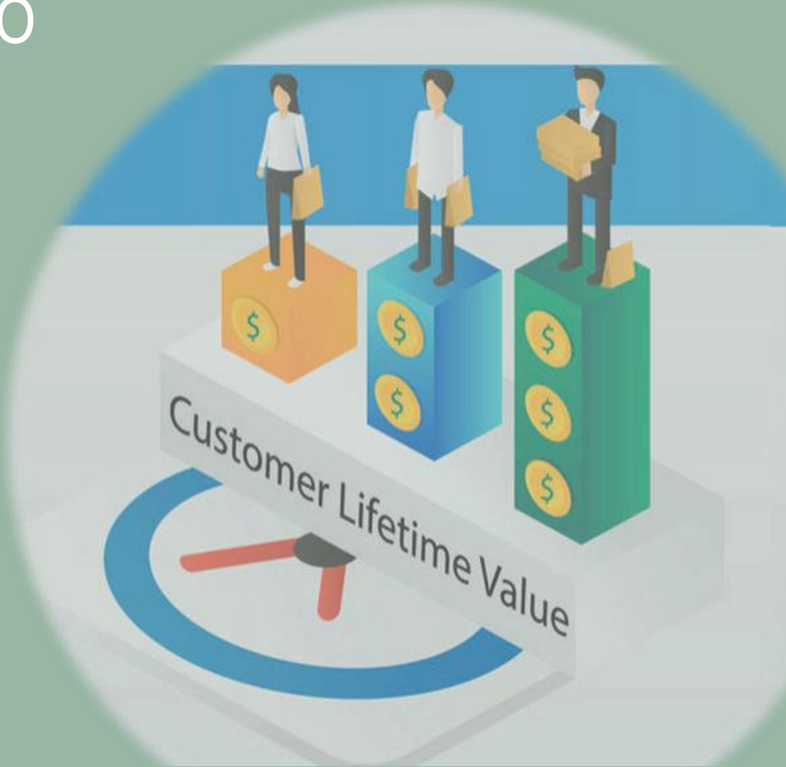
F Reference

Introduction

- Problem statement
- Definition of CLV
- RFM analysis
- Data preprocessing

Problem Statement

- Our project aims to predict the Customer Lifetime Value (CLV) for the next 3 months to identify high-value customers
- Using past transaction data from December 2010 to December 2011
- CLV = Total Transaction amount (sales_sum)

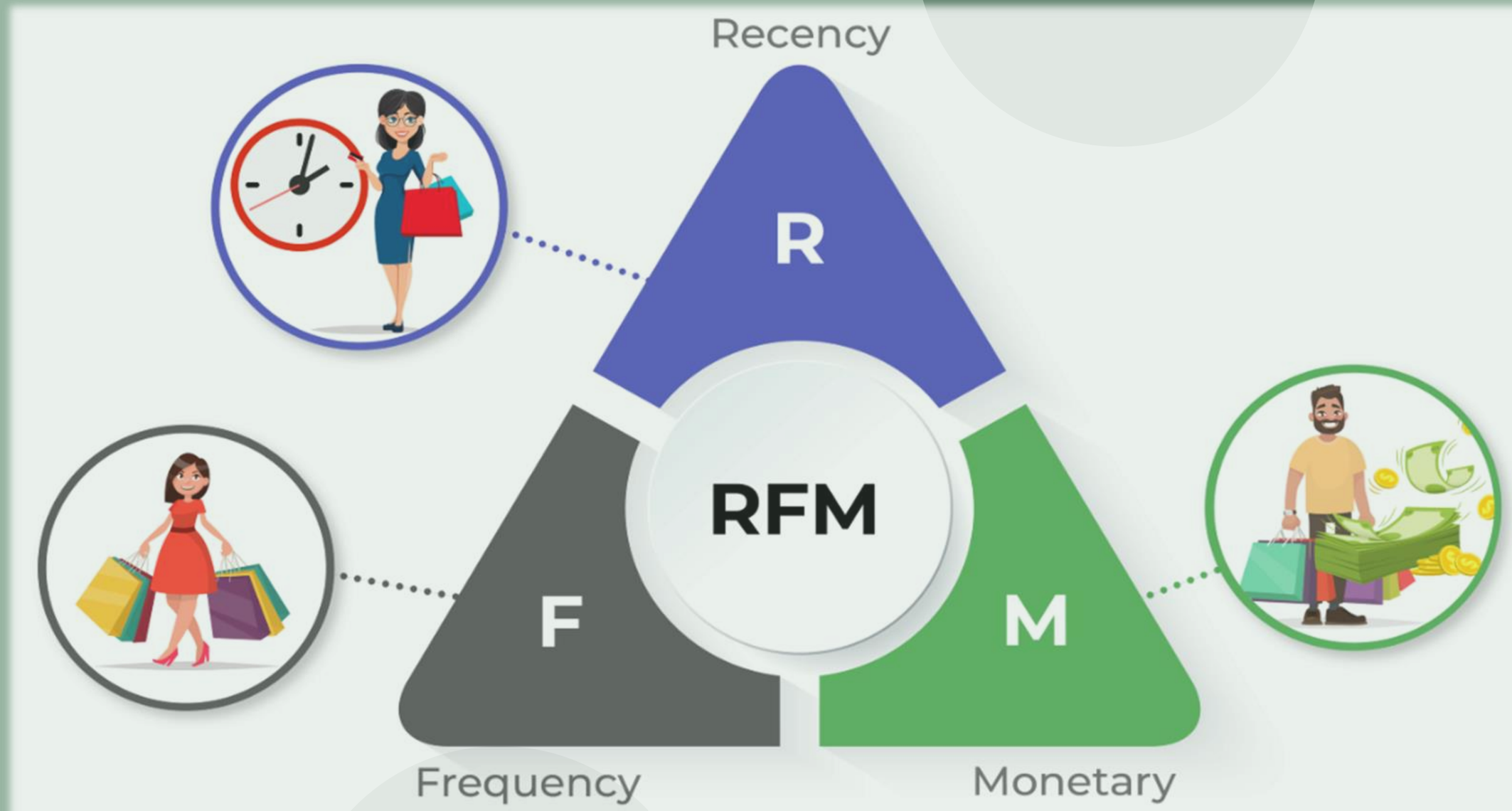


Definition of Customer Lifetime Value

- An index used to estimate the total amount of money a customer is expected to spend throughout their lifetime as a customer
- Calculated based on Monetary Value, Purchasing Frequency, and Recency
- A prediction of the net profit contributed to the whole future relationship with a customer
- The higher the CLV, the more valuable the buyer is to business

RFM analysis

	Recency (R)	Monetary value (M)	Purchase frequency (F)
Definition	How recently a customer purchased	How much a customer spend	How often a customer buy
Measurement	Average days since the last purchase	Total transaction amounts	Total number of transactions
Importance	Low recency correlates with higher customer retention	Identify high-spending customers	Show loyalty and engagement


Recency
R
RFM
F
M
Frequency
Monetary

DATA PREPOCESSING

Background of Data Set

- A dataset named Online Retail
- Comes from a non-physical online e-commerce store
- Products are mainly all-occasion gifts
- Customers are local and international wholesalers
- Contains the transactions between December, 2010 and December, 2011

Dataset Overview

- This dataset contains the transactions between December 1, 2010 and December 9, 2011
- The dimensions of the dataset are (541909, 8).

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
2	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	1/12/2010 8:26	2.55	17850	United Kingdom
3	536365	71053	WHITE METAL LANTERN	6	1/12/2010 8:26	3.39	17850	United Kingdom
4	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	1/12/2010 8:26	2.75	17850	United Kingdom
5	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	1/12/2010 8:26	3.39	17850	United Kingdom
6	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	1/12/2010 8:26	3.39	17850	United Kingdom
7	536365	22752	SET 7 BABUSHKA NESTING BOXES	2	1/12/2010 8:26	7.65	17850	United Kingdom
8	536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	1/12/2010 8:26	4.25	17850	United Kingdom
9	536366	22633	HAND WARMER UNION JACK	6	1/12/2010 8:28	1.85	17850	United Kingdom
10	536366	22632	HAND WARMER RED POLKA DOT	6	1/12/2010 8:28	1.85	17850	United Kingdom
11	536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	1/12/2010 8:34	1.69	13047	United Kingdom
12	536367	22745	POPPY'S PLAYHOUSE BEDROOM	6	1/12/2010 8:34	2.1	13047	United Kingdom
13	536367	22748	POPPY'S PLAYHOUSE KITCHEN	6	1/12/2010 8:34	2.1	13047	United Kingdom
14	536367	22749	FELTCRAFT PRINCESS CHARLOTTE DOLL	8	1/12/2010 8:34	3.75	13047	United Kingdom
15	536367	22310	IVORY KNITTED MUG COSY	6	1/12/2010 8:34	1.65	13047	United Kingdom

Data Cleaning

- 1454 null values in "Description", but this does not affect our analysis, so we will not take action on it

- 135080 null values in “CustomerID”, remove those null values

- Negative values in the "Quantity" column usually indicate returns or canceled orders

- To build an accurate predictive model, remove 10624 negative values in "Quantity" and only keep meaningful transactions for our analysis

```
# Check for null values in each column
null_counts = df.isnull().sum()
print("Null values per column:")
print(null_counts)

# Check for negative values in each numeric column
negatives = df.select_dtypes(include=['number']) < 0
negative_counts = negatives.sum()
print("\nNegative values per numeric column:")
print(negative_counts)
```

```
Null values per column:
InvoiceNo          0
StockCode          0
Description     1454
Quantity           0
InvoiceDate        0
UnitPrice          0
CustomerID    135080
Country            0
dtype: int64

Negative values per numeric column:
Quantity      10624
UnitPrice         2
CustomerID        0
dtype: int64
```

Data Cleaning

- The date range of the dataset: 2010-12-01 to 2011-12-09
- We do not have data for the entire month of December 2011
- Delete records from December 1, 2011 and later

```
# Data date range
print('Date Range: %s to %s' % (df['InvoiceDate'].min(), df['InvoiceDate'].max()))
```

```
Date Range: 2010-12-01 08:26:00 to 2011-12-09 12:50:00
```

```
# Data cleaning
# Remove those null value in customerID and negative value in Quantity
# And finally our data date range will change from 2010-12-01 to 2011-11-30

df = df.loc[df['Quantity'] > 0] # Remove quantities that are less than 0 (possibly returned items)
df = df[pd.notnull(df['CustomerID'])] # Remove blank customer IDs
df = df.loc[df['InvoiceDate'] < '2011-12-01'] # taking all of the transactions that occurred before December 01, 2011
```

Calculate sales revenue

```
# Get a orders summary dataset that shows the total in sales made per customer invoice

df['Sales'] = df['Quantity'] * df['UnitPrice'] # Create a Sales Revenue Column

orders_df = df.groupby(['CustomerID', 'InvoiceNo']).agg({'Sales': sum,'InvoiceDate': max})
orders_df.head(10)
```

CustomerID	InvoiceNo	Sales	InvoiceDate
12346.0	541431	77183.60	2011-01-18 10:01:00
12347.0	537626	711.79	2010-12-07 14:57:00
	542237	475.39	2011-01-26 14:30:00
	549222	636.25	2011-04-07 10:43:00
	556201	382.52	2011-06-09 13:01:00
	562032	584.91	2011-08-02 08:48:00
	573511	1294.32	2011-10-31 12:25:00
12348.0	539318	892.80	2010-12-16 19:09:00
	541998	227.44	2011-01-25 10:42:00
	548955	367.00	2011-04-05 10:47:00

Sales revenue:

Quantity × UnitPrice

Creating the Sales column is necessary to calculate the total revenue for each invoice

Original solution

- Original solution
  - Data preparation
  - Machine learning
- Limitations & Motivations

Machine Learning

Step 01:

Divided known data into training set (70%) and test set (30%)

[Finished in data preparation]

Step 02:

Use Linear Regression to fit training data

Step 03:

Mach features and coefficients into a Data Frame

Step 04:

Use training set to predict test set

Step 05:

Use R^2 and MAE to evaluate performance of model

Data Preparation

Time Series Analysis

Dividing time into 5 consecutive periods and using them to train a machine learning model after (3 months for each)

M1
M2
M3
M4
M5

RFM Analysis

Selecting & Counting sales average, sum, and count for each period to predict customer lifetime value

R(ecency): Recent time of purchase -> Each period

F(rquency): Number of purchases -> Sales count

M(onetary Value): Amount of purchase
-> Sales average and sum

Data Preparation (Before machine learning)

3 variables

Sales sum = Sales count × Sales average

Training set: M2 – M5

Testing set: M1

	sales_avg_M_2	sales_avg_M_3	sales_avg_M_4	sales_avg_M_5	sales_count_M_2	sales_count_M_3	sales_count_M_4	sales_count_M_5	sales_sum_M_2	sales_sum_M_3	sales_sum_M_4	sales_sum_M_5	CustomerID	CLV_3M
NaN	0.000000	0.000	77183.600000	0.00	0.0	0.0	1.0	0.0	0.00	0.00	77183.60	0.00	12346.0	0.00
5.0	584.910000	509.385	475.390000	711.79	1.0	2.0	1.0	1.0	584.91	1018.77	475.39	711.79	12347.0	1294.32
NaN	310.000000	367.000	227.440000	892.80	1.0	1.0	1.0	1.0	310.00	367.00	227.44	892.80	12348.0	0.00
NaN	0.000000	0.000	334.400000	0.00	0.0	0.0	1.0	0.0	0.00	0.00	334.40	0.00	12350.0	0.00
14.0	316.250000	0.000	312.362000	0.00	2.0	0.0	5.0	0.0	632.50	0.00	1561.81	0.00	12352.0	311.73
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
NaN	0.000000	0.000	180.600000	0.00	0.0	0.0	1.0	0.0	0.00	0.00	180.60	0.00	18280.0	0.00
NaN	0.000000	80.820	0.000000	0.00	0.0	1.0	0.0	0.0	0.00	80.82	0.00	0.00	18281.0	0.00
NaN	100.210000	0.000	0.000000	0.00	1.0	0.0	0.0	0.0	100.21	0.00	0.00	0.00	18282.0	0.00
9217.0	92.696667	131.170	105.966667	0.00	3.0	4.0	3.0	0.0	278.09	524.68	317.90	0.00	18283.0	766.21
9219.0	0.000000	765.280	0.000000	0.00	0.0	1.0	0.0	0.0	0.00	765.28	0.00	0.00	18287.0	1072.00

Sales AVERAGE of M2 to M5

Sales COUNT of M2 to M5

Sales SUM of M2 to M5

Sales SUM of M1

Step 02 & 03: Linear Regression

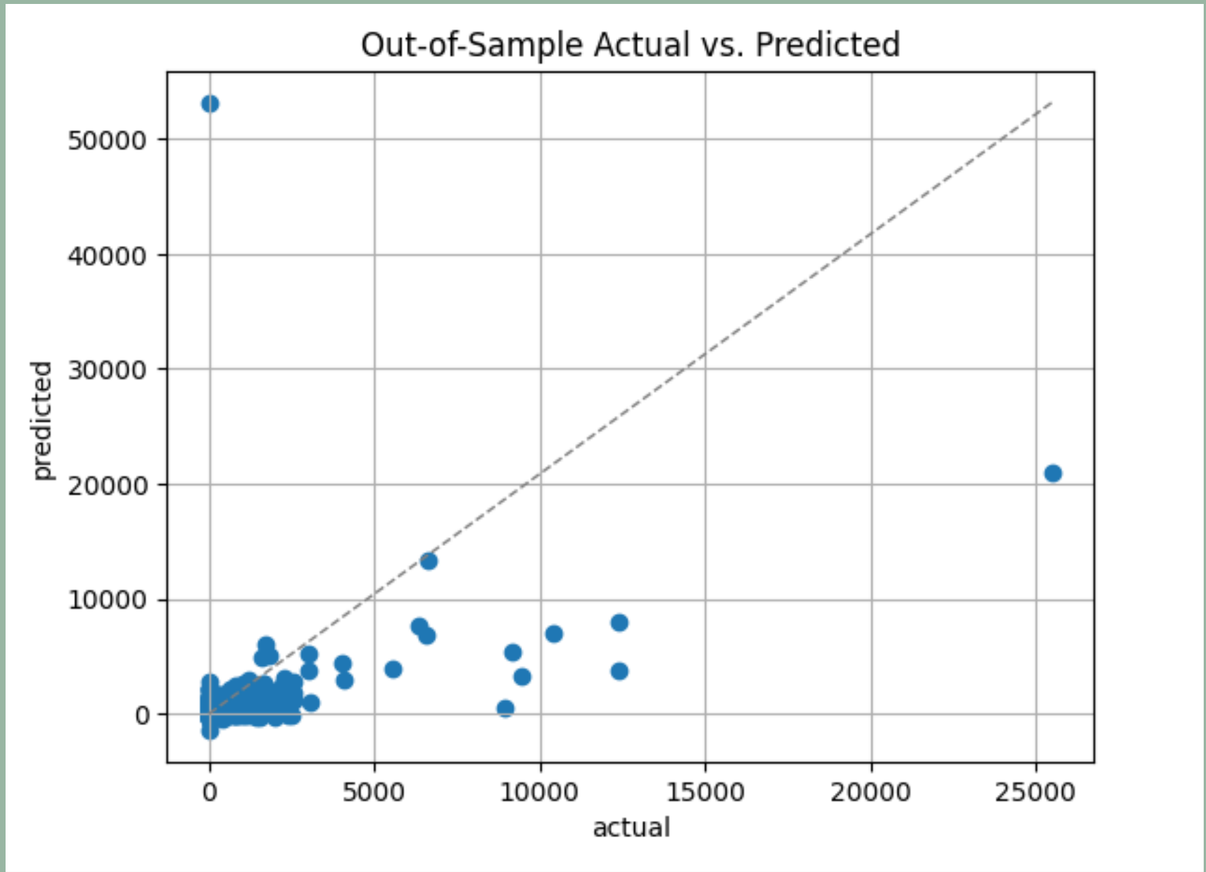
Definition: Minimize error between predicted value and actual value by finding a best-fit straight-line

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \epsilon$$

y = predict value (CLV)
β_1 to β_n = Coefficient (impact of each feature)
x_1 to x_n = Feature variables (sales average, sum, and count)

	feature	coef
0	sales_avg_M_2	0.271605
1	sales_avg_M_3	-0.788028
2	sales_avg_M_4	0.520973
3	sales_avg_M_5	-0.430464
4	sales_count_M_2	115.941739
5	sales_count_M_3	30.196844
6	sales_count_M_4	-169.169063
7	sales_count_M_5	-33.519652
8	sales_sum_M_2	0.191658
9	sales_sum_M_3	0.402093
10	sales_sum_M_4	0.169278
11	sales_sum_M_5	0.839973

Step 04: Visualization



Scatter plot of actual value and predicted value

A gray dashed line represents the ideal situation (Predicted value = Actual value)

- The closer to the line and the points, the more accurate the model's prediction

- If the distribution of points deviates far from the line (especially in high or low areas), the model has larger prediction errors

Step 05: R^2 and MAE

R^2 (Coefficient of determination)

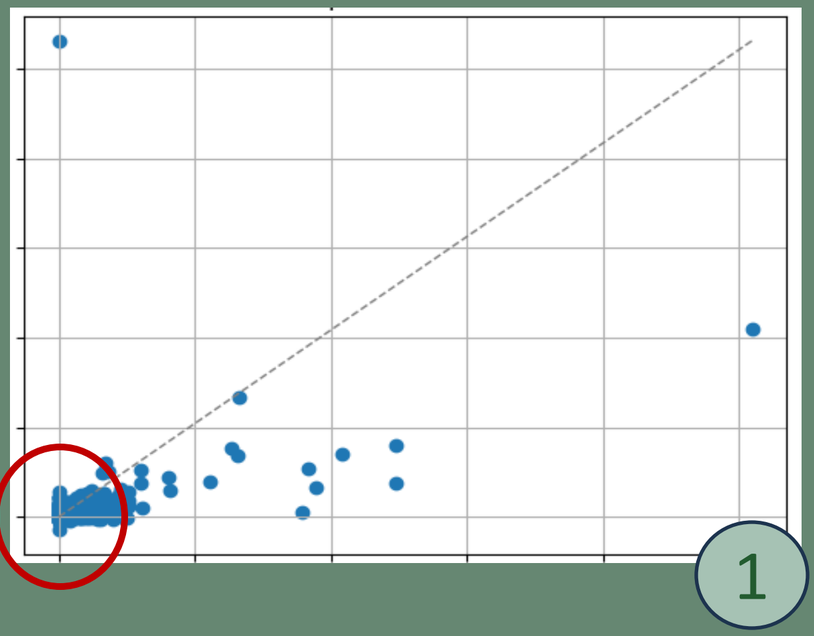
- Measures the goodness of the model to generalize new data
- Closer to 1: good generalization
- Negative: catastrophic failure

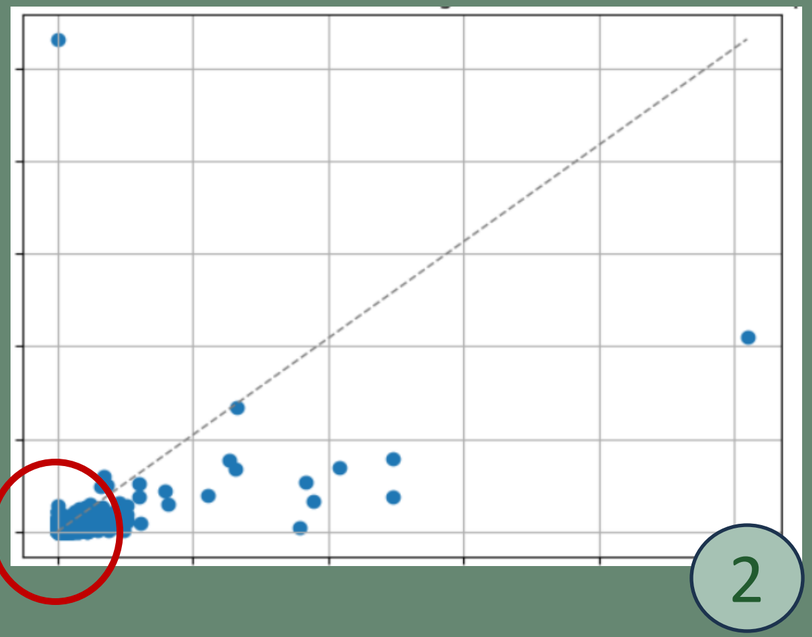
MAE (Median absolute error)

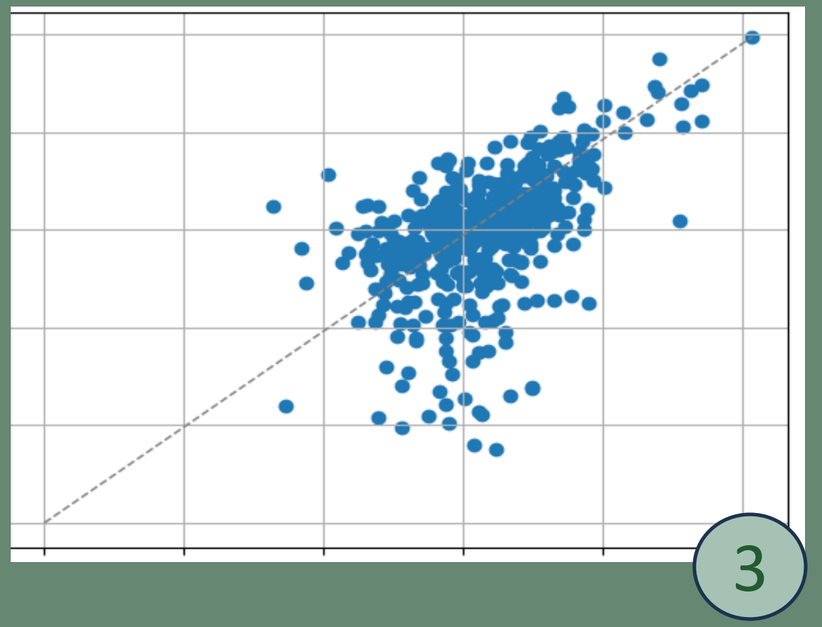
- Measures the prediction accuracy of the model
- The smaller the median, the better the prediction

$$R^2_{OS} = 1 - \frac{\sum_{t=1}^{T}(r_t - \widehat{r}_t)^2}{\sum_{t=1}^{T}(r_t - \overline{r}_t)^2},$$

$$\mathbf{MedAE} = \mathbf{median}(|Y_i - \hat{Y}_i|)$$





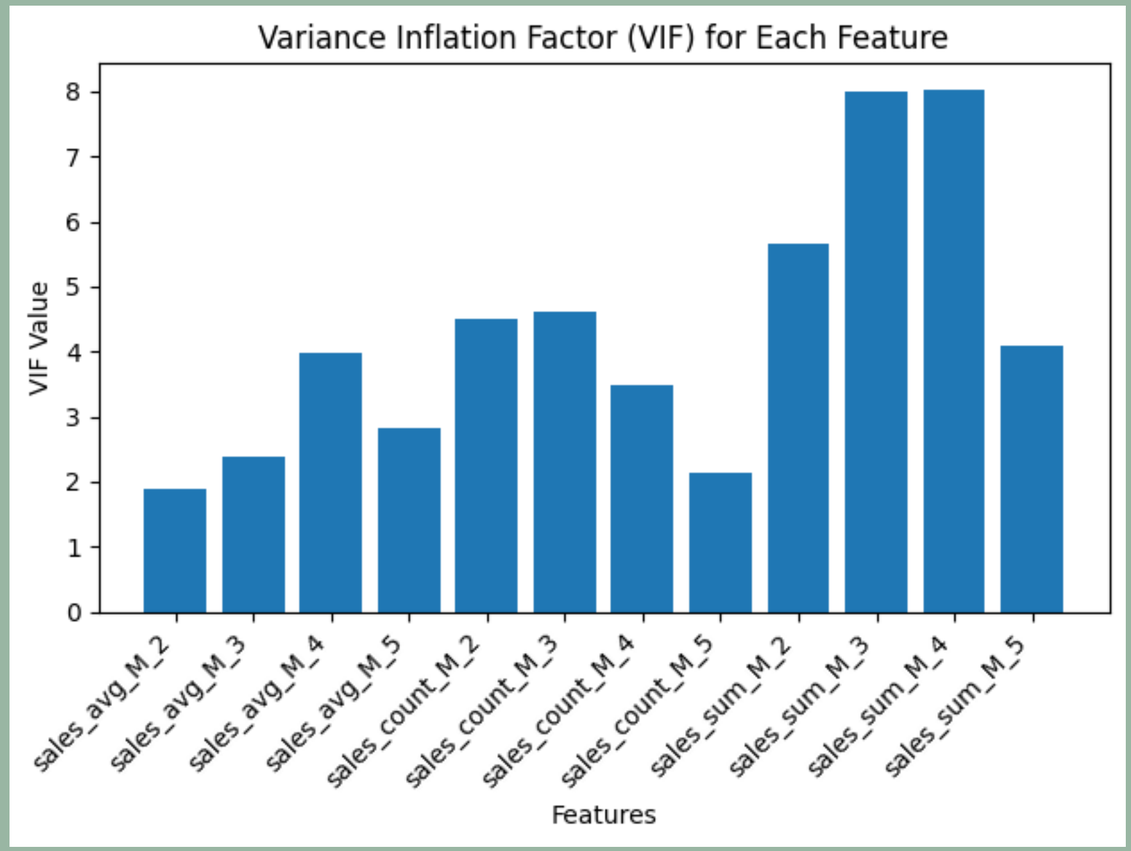


	Change	Distribution	R^2	MAE
Plot 1	Original Plot	Skewed distribution	-1.0251	228.7129
Plot 2	Remove negative CLV	Skewed distribution	-1.0532	250.6356
Plot 3	Remove negative & 0 CLV Log transformation	Normal distribution (without outliers)	-0.4497	0.5542

Limitations and Motivations

1. Dependence Between Parameters in Linear Regression

- Using sales AVERAGEs, sales COUNTs and sales SUMs as parameters.
- $AVERAGE \times COUNT = SUM$
- Linear relationships⟶ **Multicollinearity**
  - High VIFs for sales_sum
    - Unstable coefficient estimates
    - Reduce the predictive power.



Data Preparation for Linear Regression

3 variables

Sales sum = Sales count × Sales average

Training set: M2 – M5

Testing set: M1

	sales_avg_M_2	sales_avg_M_3	sales_avg_M_4	sales_avg_M_5	sales_count_M_2	sales_count_M_3	sales_count_M_4	sales_count_M_5	sales_sum_M_2	sales_sum_M_3	sales_sum_M_4	sales_sum_M_5	CustomerID	CLV_3M
NaN	0.000000	0.000	77183.600000	0.00	0.0	0.0	1.0	0.0	0.00	0.00	77183.60	0.00	12346.0	0.00
5.0	584.910000	509.385	475.390000	711.79	1.0	2.0	1.0	1.0	584.91	1018.77	475.39	711.79	12347.0	1294.32
NaN	310.000000	367.000	227.440000	892.80	1.0	1.0	1.0	1.0	310.00	367.00	227.44	892.80	12348.0	0.00
NaN	0.000000	0.000	334.400000	0.00	0.0	0.0	1.0	0.0	0.00	0.00	334.40	0.00	12350.0	0.00
14.0	316.250000	0.000	312.362000	0.00	2.0	0.0	5.0	0.0	632.50	0.00	1561.81	0.00	12352.0	311.73
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
NaN	0.000000	0.000	180.600000	0.00	0.0	0.0	1.0	0.0	0.00	0.00	180.60	0.00	18280.0	0.00
NaN	0.000000	80.820	0.000000	0.00	0.0	1.0	0.0	0.0	0.00	80.82	0.00	0.00	18281.0	0.00
NaN	100.210000	0.000	0.000000	0.00	1.0	0.0	0.0	0.0	100.21	0.00	0.00	0.00	18282.0	0.00
9217.0	92.696667	131.170	105.966667	0.00	3.0	4.0	3.0	0.0	278.09	524.68	317.90	0.00	18283.0	766.21
9219.0	0.000000	765.280	0.000000	0.00	0.0	1.0	0.0	0.0	0.00	765.28	0.00	0.00	18287.0	1072.00

Sales AVERAGE of M2 to M5

Sales COUNT of M2 to M5

Sales SUM of M2 to M5

Sales SUM of M1

Limitations and Motivations

2. Complexity of data process completeness

Linear Regression

- Splitting the data into M_1, M_2, M_3, M_4, M_5
- Finding MEAN, COUNT, and SUM of sales for each period manually
- **TOO MANY STEP**

BG/NBG with Gamma Gamma

- Using packages to calculate key metrics (frequency, recency, and monetary value) from raw transactional data
- **EASIER**

Limitations and Motivations

3. Results of Prediction

Linear Regression

- The Model is trained for ONE output → **sales_sum in M_1**
- **One model, One output**

BG/NBG with Gamma Gamma

- Provide a more comprehensive view of customer behaviour
- For instance, purchasing frequency

Limitations and Motivations

4. Negative CLV predictions in Linear Regression model

- Including negative value in predicted M_1 Customer Lifetime Value
- CLV = Total Transaction amount (sales_sum)

Motivate us to find another model that would not give negative values

Limitations and Motivations

5. Not providing future information to find CLV

- Original solution does not provide any future predictions
- Unable to identify the most valuable customer, which is the primary purpose of conducting this prediction.

Extension 1

- New Proposed Model:
  BG/NBD with Gamma Gamma model
- Compare Extension and original solution

BG/NBD with Gamma Gamma model

Step 01:
Data prepare for fitting the model

Step 02:
Fitting BG/NBD Model

Step 03:
Fitting Gamma Gamm model

Step 04:
Predicting each customer CLV in future 3 months

Further:
Compare Extension and original solution

Predicted CLV	Predicted frequency × Predicted monetary value
BG/NBD model	Predict how often customers will buy in the next 90 days (predicted frequency)
Gamma gamma model	Estimates how much customers will spend each time they buy **(predicted *monetary value) *average purchase value per transaction**

S1: Data prepare for fitting the model

	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

The required library and function:

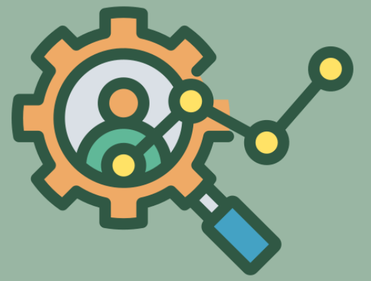
```
!pip install lifetimes
from lifetimes import BetaGeoFitter
from lifetimes import GammaGammaFitter
from lifetimes.utils import summary_data_from_transaction_data
from lifetimes.plotting import plot_period_transactions
from lifetimes.plotting import plot_frequency_recency_matrix
```

```
df_rfmt = summary_data_from_transaction_data(transactions = df,
                                             customer_id_col = 'CustomerID',
                                             datetime_col = 'InvoiceDate',
                                             monetary_value_col = 'Sales')
df_rfmt.head()
```

CustomerID	frequency	recency	T	monetary_value
12346.0	0.0	0.0	325.0	0.000000
12347.0	6.0	365.0	367.0	599.701667
12348.0	3.0	283.0	358.0	301.480000
12349.0	0.0	0.0	18.0	0.000000
12350.0	0.0	0.0	310.0	0.000000

S2: Fitting BG/NBD Model

how often customers will buy in the next 90 days?



```
model = BetaGeoFitter(0)
model.fit(df_rfmt['frequency'],
          df_rfmt['recency'],
          df_rfmt['T'])
model.summary
```

	coef	se(coef)	lower
r	0.826433	0.026780	
alpha	68.890678	2.611055	
a	0.003443	0.010347	
b	6.749363	22.412933	

Distribution	Parameter	Description	In our case
Gamma (Models the frequency of all customer purchases)	r	Models variance of purchase frequency	Relative low r, Some buy often, while others buy very little
	α	Controls all the customer's average buying frequency	Relative High **α**, Customers buy less frequently
Beta(Models the probability that a customer is still alive)	a	Represents how quickly customers stop buying	Relative Low **a**, Customers are unlikely to loss quickly
	b	Represents how likely customers are to stay active	Relative High **b**, Most customers are likely to stay active

S2: Fitting BG/NBD Model

how often customers will buy in the next 90 days?

```
# Predicting the number of purchases in the next 90 days for all customers.
df_rfmt['predicted_purchases'] = model.conditional_expected_number_of_purchases_up_to_time(90,
                                                                                           df_rfmt['frequency'],
                                                                                           df_rfmt['recency'],
                                                                                           df_rfmt['T'])


# Droped the NA predicted_purchases
# The NA may occur if any of the input columns (frequency, recency, or T) have NA values, the model cannot make a prediction
df_rfmt.dropna(subset=['predicted_purchases'], inplace=True)
df_rfmt
```

CustomerID	frequency	recency	T	monetary_value	predicted_purchases
12346.0	0.0	0.0	325.0	0.000000	0.188812
12347.0	6.0	365.0	367.0	599.701667	1.408759
12348.0	3.0	283.0	358.0	301.480000	0.805911
12349.0	0.0	0.0	18.0	0.000000	0.855637
12350.0	0.0	0.0	310.0	0.000000	0.196286

The number of predicted purchases will then used to get the predicted CLV.

S3: Fitting Gamma Gamm model

How much will they spend each time?

Gamma-Gamma model is used to predict how much customers spend on average.

```
df_rfmt = df_rfmt[df_rfmt['monetary_value'] > 0]
gg_model = GammaGammaFitter()
gg_model.fit(df_rfmt['frequency'], df_rfmt['monetary_value']).summary
```

	coef	se(coef)	lower 95% bound	upper 95% bound
p	2.103523	0.111998	1.884007	2.323039
q	3.449907	0.139042	3.177385	3.722429
v	485.570938	42.595555	402.083650	569.058225

Parameter	Description
P	Describes variance of individual customers monetary value
q	Describes variance of monetary value across all customer
V	Controls the average monetary value across all customers

Customer with only 1 transaction number are removed (same with original solution)

S3: Fitting Gamma Gamm model

How much will they spend each time?

```
df_rfmt['pred_monetary'] = gg_model.conditional_expected_average_profit(
        df_rfmt['frequency'],
        df_rfmt['monetary_value'])
df_rfmt
```

CustomerID	frequency	recency	T	monetary_value	predicted_purchases	pred_monetary
12347.0	6.0	365.0	367.0	599.701667	1.408759	569.988807
12348.0	3.0	283.0	358.0	301.480000	0.805911	333.762672
12352.0	6.0	260.0	296.0	368.256667	1.682315	376.166864
12356.0	2.0	303.0	325.0	269.905000	0.645367	324.008941
12358.0	1.0	149.0	150.0	683.200000	0.750395	539.930643
...	...	...	...	...	...	...
18272.0	5.0	244.0	246.0	487.752000	1.664270	474.369525
18273.0	2.0	255.0	257.0	76.500000	0.780057	201.781295
18282.0	1.0	119.0	126.0	77.840000	0.842753	260.275833
18283.0	13.0	334.0	337.0	152.802308	3.064342	174.518797
18287.0	2.0	159.0	201.0	536.000000	0.941627	492.175045

S4: Predicting each customer CLV in future 3months

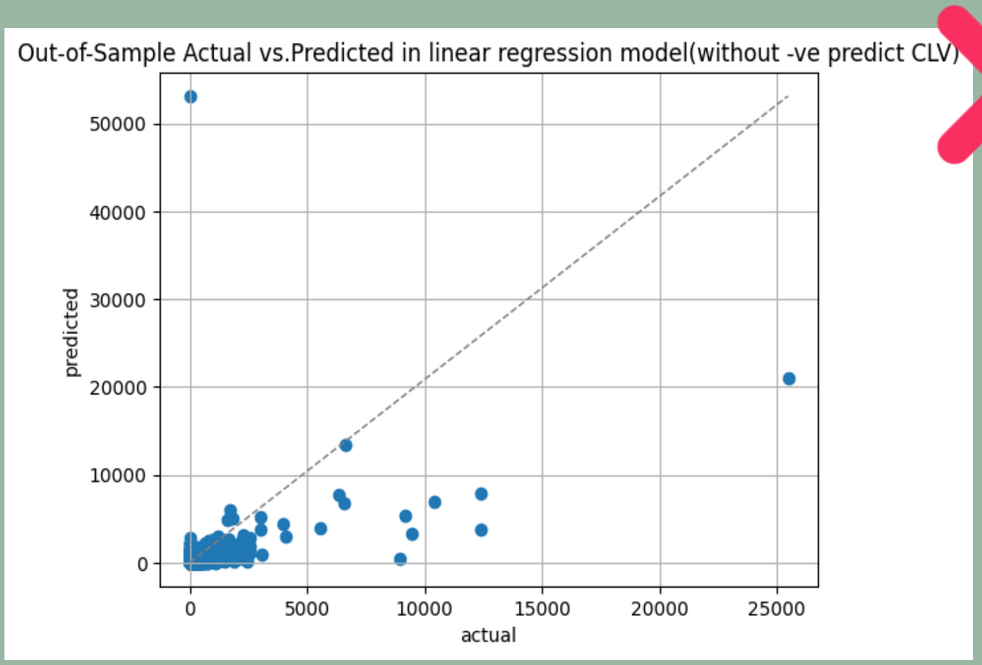
```
df_rfmt['pred_3month_CLV'] = df_rfmt['predicted_purchases']*df_rfmt['pred_monetary']
df_rfmt
```

CustomerID	frequency	recency	T	monetary_value	predicted_purchases	pred_monetary	pred_3month_CLV
12347.0	6.0	365.0	367.0	599.701667	1.408759	569.988807	802.976798
12348.0	3.0	283.0	358.0	301.480000	0.805911	333.762672	268.982959
12352.0	6.0	260.0	296.0	368.256667	1.682315	376.166864	632.831326
12356.0	2.0	303.0	325.0	269.905000	0.645367	324.008941	209.104598
12358.0	1.0	149.0	150.0	683.200000	0.750395	539.930643	405.161078
...	...	...	...	...	...	...	...
18272.0	5.0	244.0	246.0	487.752000	1.664270	474.369525	789.478815
18273.0	2.0	255.0	257.0	76.500000	0.780057	201.781295	157.400999
18282.0	1.0	119.0	126.0	77.840000	0.842753	260.275833	219.348197
18283.0	13.0	334.0	337.0	152.802308	3.064342	174.518797	534.785352
18287.0	2.0	159.0	201.0	536.000000	0.941627	492.175045	463.445170

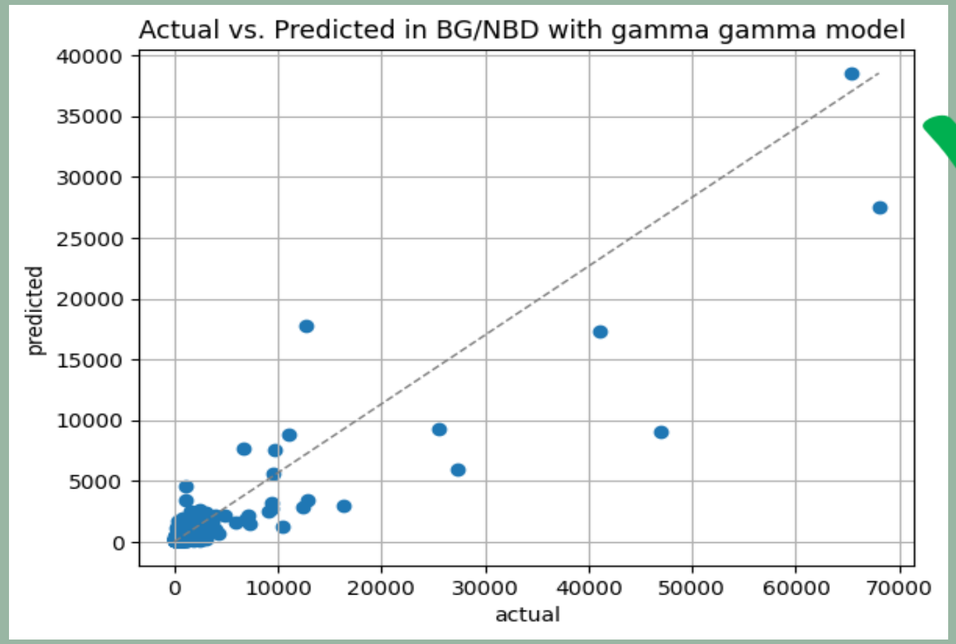
Predicted CLV = Predicted frequency × Predicted monetary value.

The Predictions are accurate?

- Use the same period with the features in original solution to fit the model
- Predicted customer CLV in the M1 period



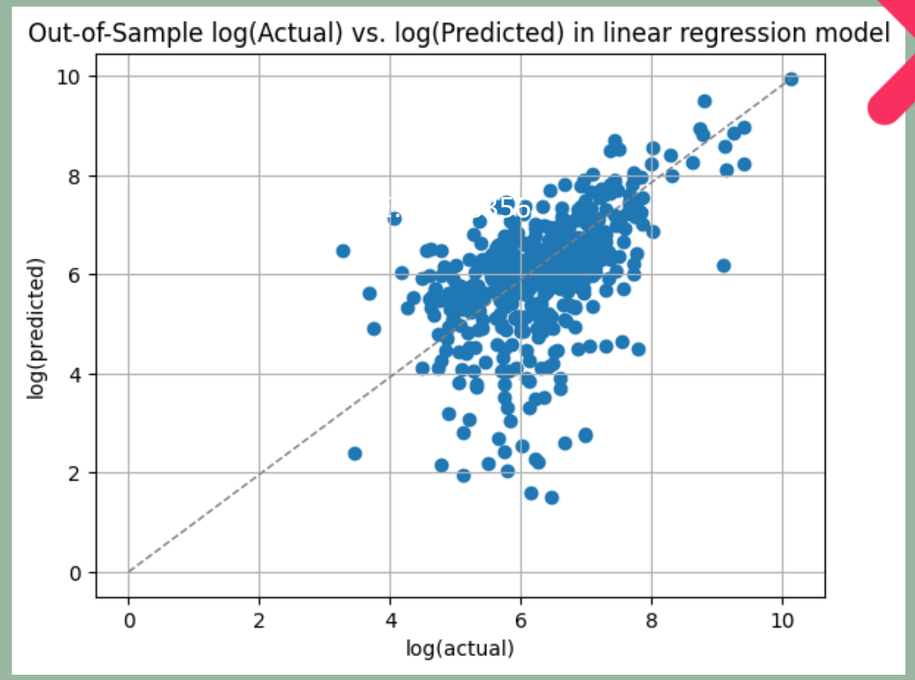
MAE: 250.6356
Number of data points: 873



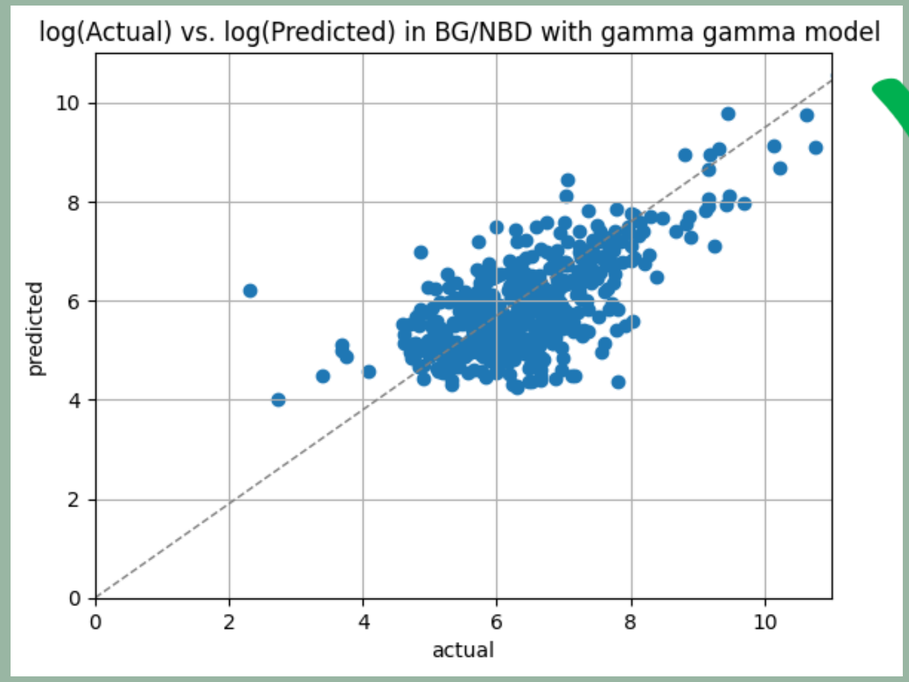
MAE: 229.4651
Number of data points: 873

The Predictions are accurate?

After removing those actual CLV value = 0 and take log



Linear regression
In original solution



BG/NBD with gamma gamma
In extension solution

For someone who interested about

1. The assumptions and fundamental principles of the BG/NBD model

Fader, P. S., Hardie, B. G., & Lee, K. L. (2005). “Counting your customers” the easy way: An alternative to the Pareto/NBD model. Marketing Science, 24(2), 275–284. https://brucehardie.com/papers/018/fader_et_al_mksc_05.pdf

2. The details of function in the lifetime package

lifetimes 0.11.2 documentation. (n.d.). https://lifetimes.readthedocs.io/en/latest/lifetimes.html

Extension 2

- K-means clustering

K-Means Clustering for Customer Segmentation

Step 01:
Determination of Optimal Cluster Number (k)

Step 02:
Clustering and Analysis

Step 03:
Identification of High-Value Customer Segments

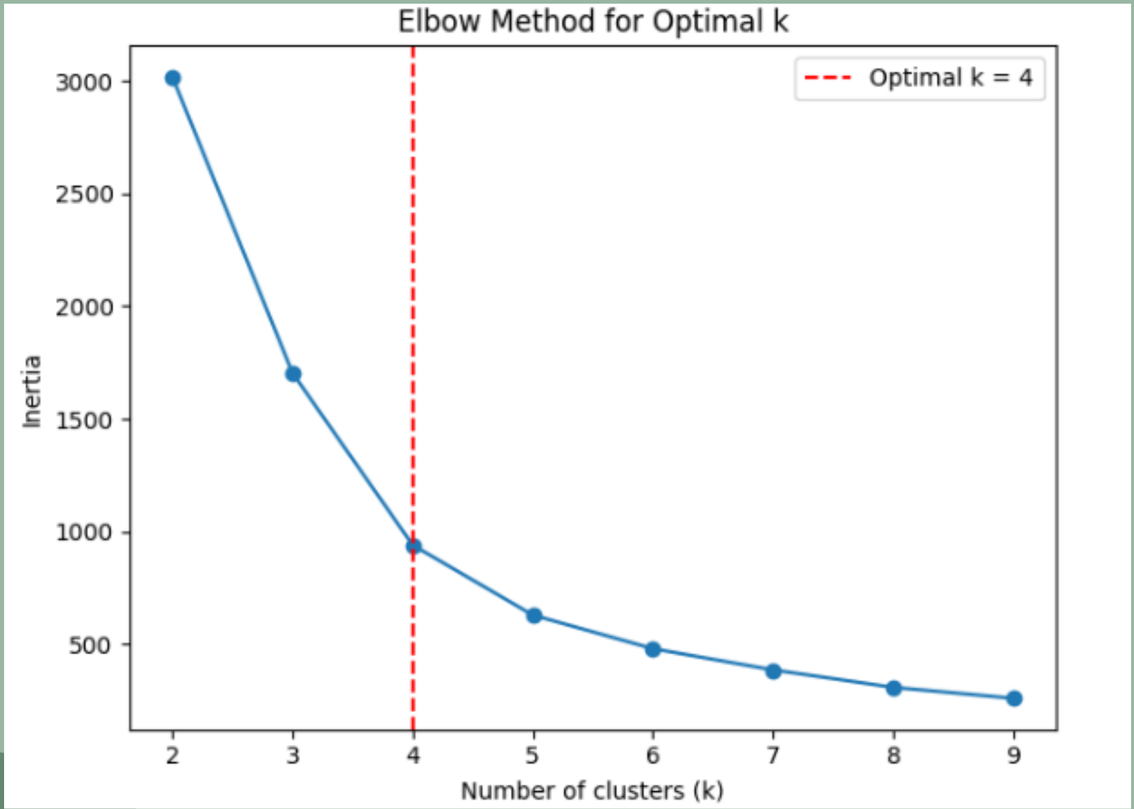
Step 04:
Visualization

Further:
Compare Extension and original solution

Principle of K-Means Clustering

1. Randomly select K data points as initial centroids
2. Assign each data point to the nearest centroid
3. Recalculate the centroids as the mean of all points assigned to each cluster
4. Repeat the assignment and update steps until convergence (when centroids no longer change significantly)

S1: Determination of Optimal Cluster Number (k)



1. The Elbow Method is utilized to determine the optimal number of clusters
2. The optimal number of clusters was identified using the KneeLocator based on the elbow point

S2: Clustering and Analysis

Cluster Summary:

	cluster	avg_predicted_purchases	avg_pred_monetary	count	total_value
0	1	0.599739	78051.285456	1	46810.366377
1	3	19.590359	516.419579	7	10116.844977
2	2	3.859390	618.981990	307	2388.893016
3	0	1.078574	402.911190	2475	434.569454

1. Applied K-means clustering with the optimal k value

2. Computed cluster centers and summarized each cluster's average predicted purchases, average predicted monetary value, and total customer count

3. Sorted clusters based on their total predicted value to highlight the most valuable groups

S3: Identification of High-Value Customer Segments

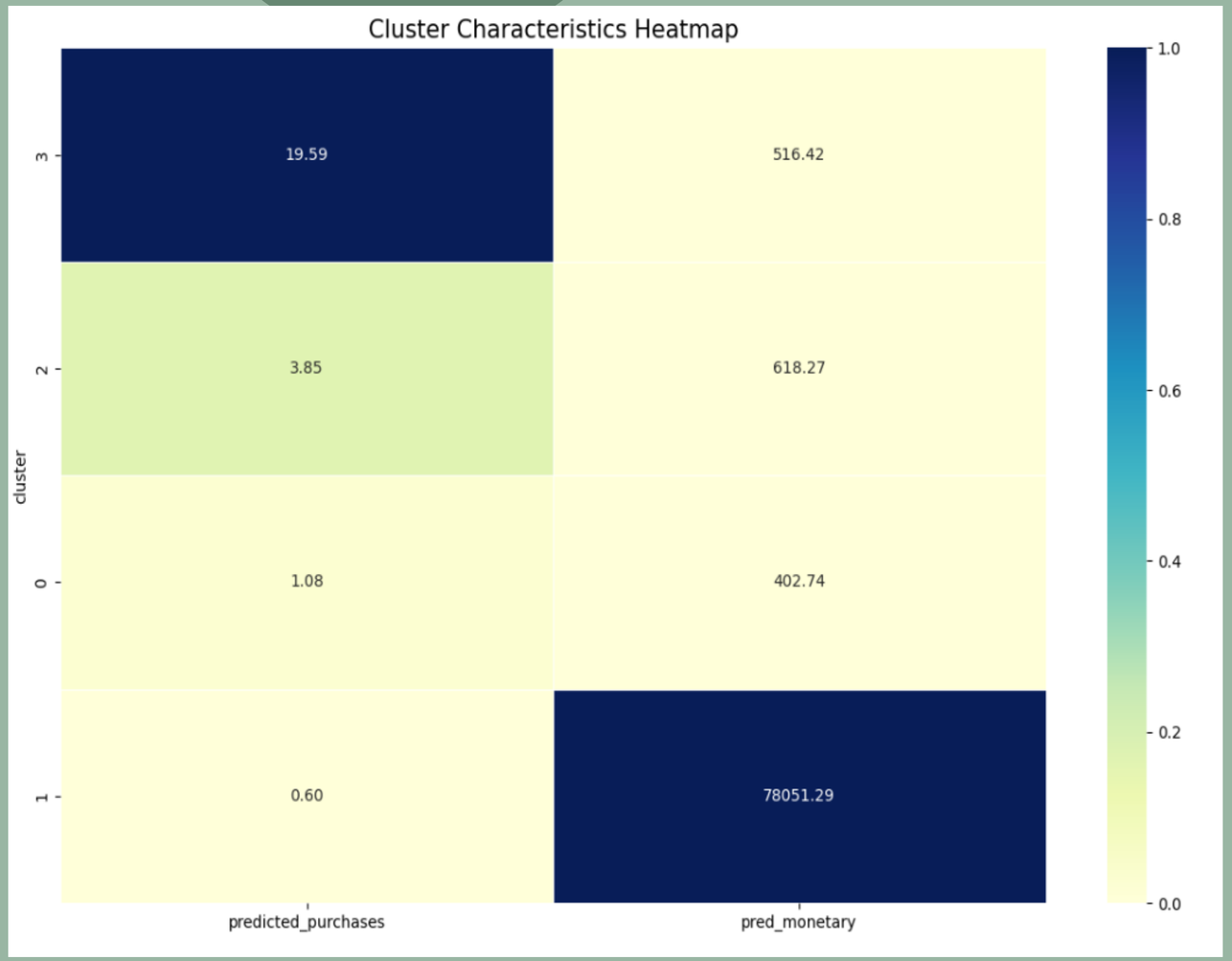
High-value customer IDs:

CustomerID	frequency	recency	T	monetary_value	predicted_purchases	\
12748.0	113.0	373.0	373.0	298.360885	23.175061	
12971.0	70.0	369.0	372.0	159.211286	14.452417	
13089.0	65.0	367.0	369.0	893.714308	13.524270	
14606.0	88.0	372.0	373.0	135.890114	18.084835	
14911.0	131.0	372.0	373.0	1093.661679	26.839684	
15311.0	89.0	373.0	373.0	677.729438	18.288586	
16446.0	1.0	205.0	205.0	168469.600000	0.599739	
17841.0	111.0	372.0	373.0	364.452162	22.767661	

1. Selected the top 2 high-value clusters based on average predicted purchases and average predicted monetary values.

2. Extracted customer IDs from these high-value clusters for targeted marketing or personalized strategies.

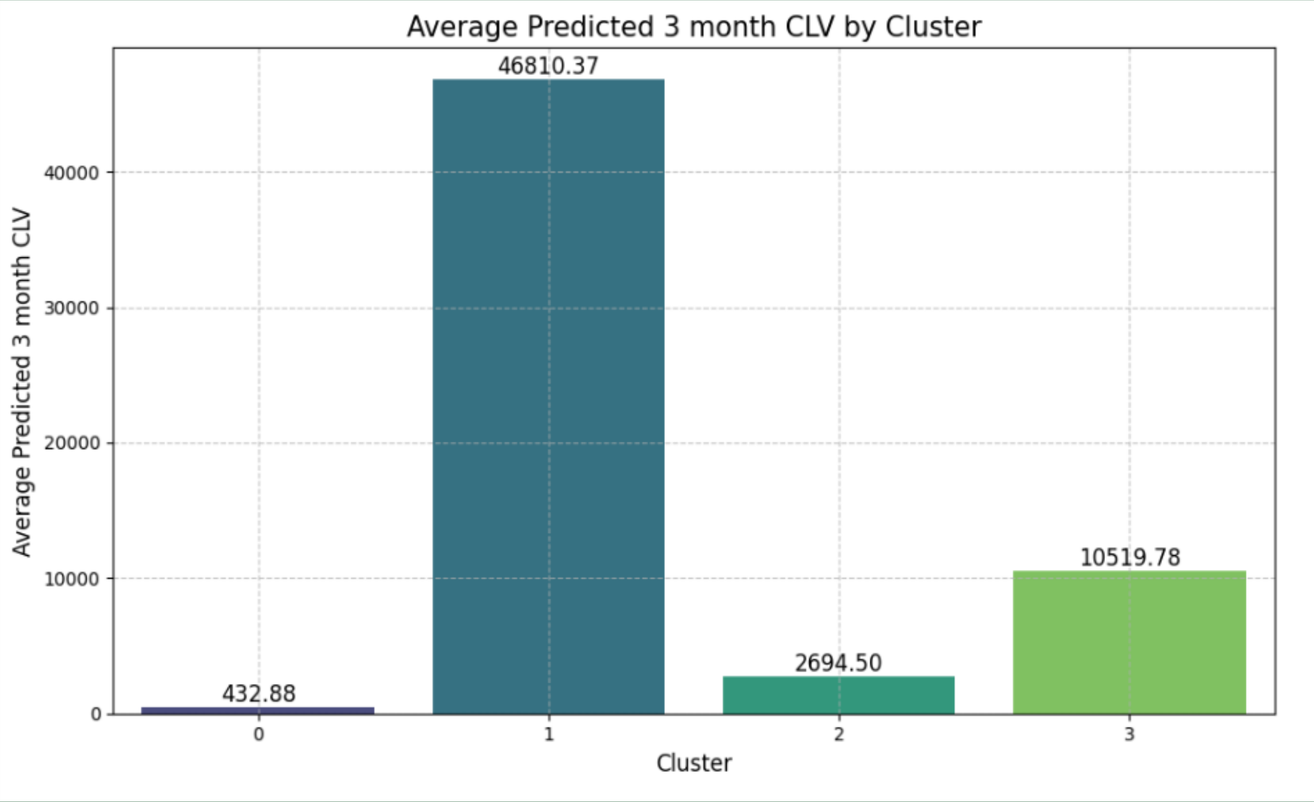
S4: Visualization



1. Visualized the clustering results using a scatter plot, highlighting cluster centers and high-value clusters

2. Created a heatmap to illustrate the distinguishing characteristics of each cluster

Further Work:
Calculate average predicted CLV for each cluster



The average predicted CLV of Cluster1 is significantly higher than those of other clusters

Conclusion

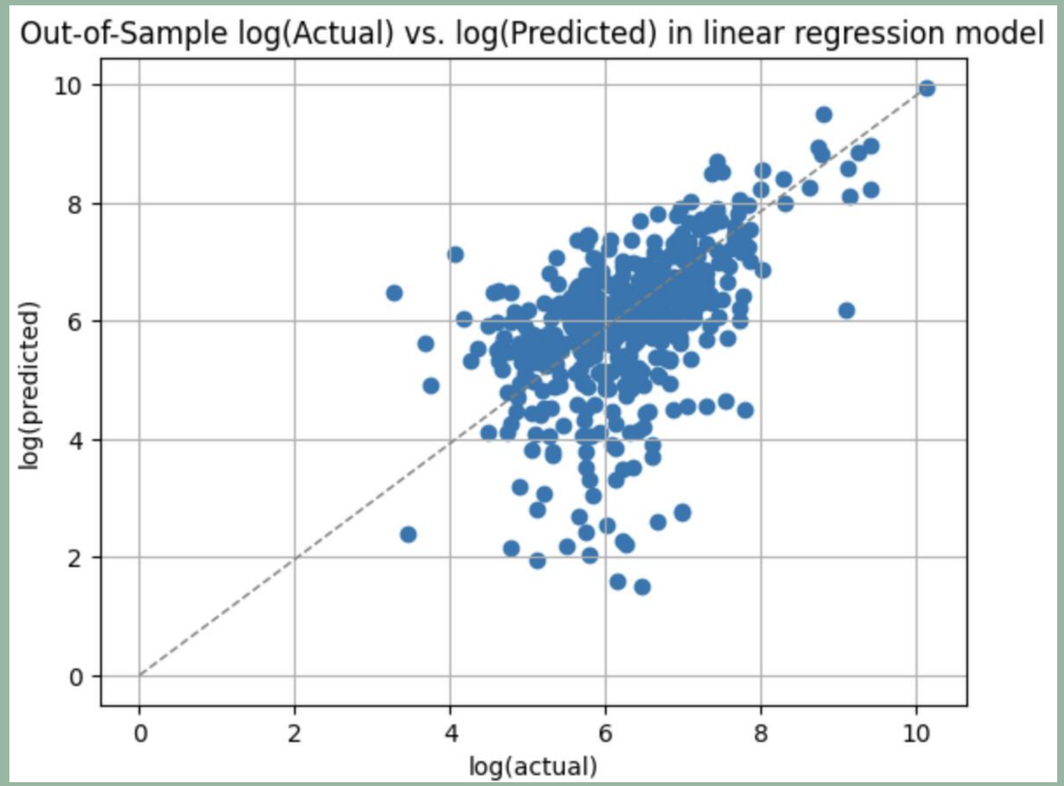
Conclusion

Purpose: finding the targeted customer group

Original solution: using linear model to predict the CLV in future three months

Drawback:

1. Not vary accurate (since there is a linear relationship between parameters)
2. Complexity of data process completeness
3. Negative CLV predictions in Linear Regression model
4. Not providing future information to find CLV

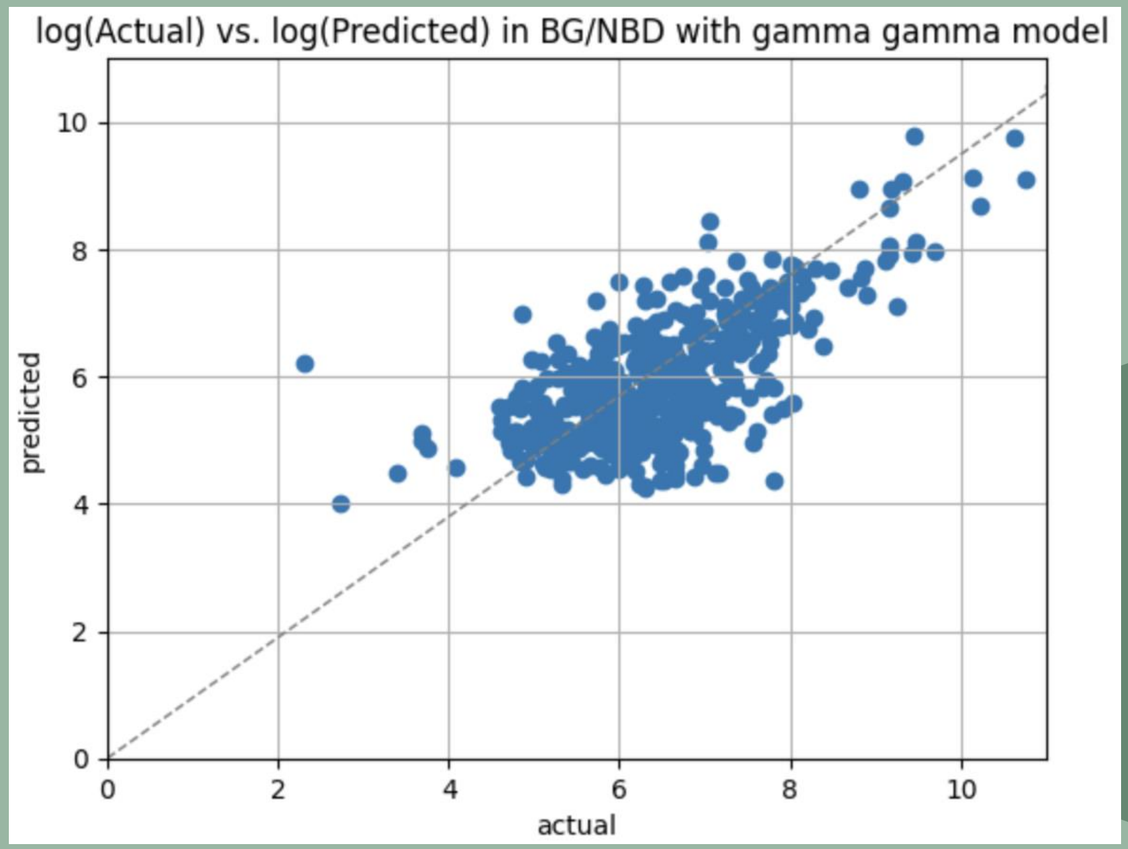


Extension 1:

Using BG/NBD & gamma-gamma model to predict the frequency and monetary value in future three months

Advantage:

1. More accurate (after comparing with linear model)
2. More flexible (since the linear model can only predict the future CLV monthly, but BGNBD & gamma-gamma can predict the future CLV daily)



Extension 2:

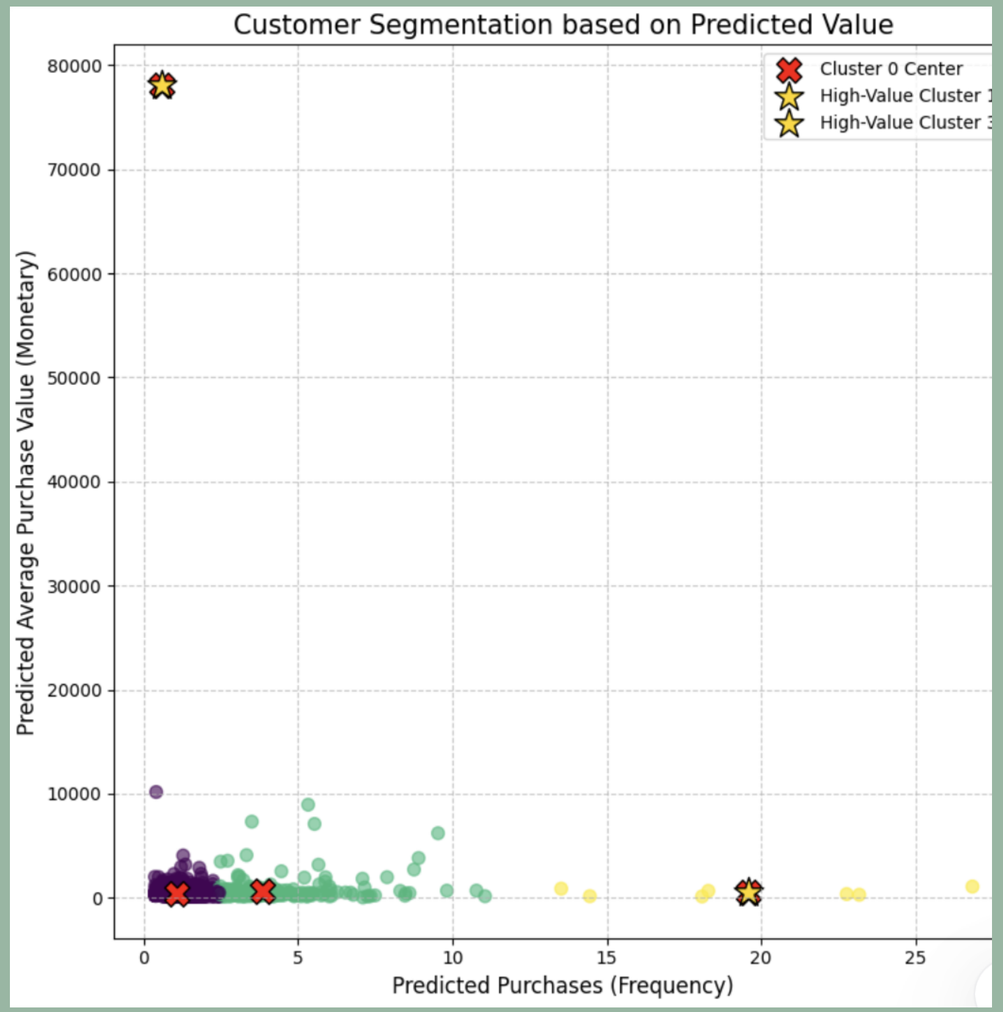
Using K-means clustering to find the targeted customer group

Advantage:

Giving the customer IDs of the targeted customer group directly

CustomerID	pred_monetary	pred_3month_CLV	cluster
12748.0	299.570360	6942.561479	3
12971.0	163.428864	2361.942010	3
13089.0	885.321458	11973.326310	3
14606.0	139.560895	2523.935709	3
14911.0	1087.698039	29193.471295	3
15311.0	674.360499	12333.099858	3
16446.0	78051.285456	46810.366377	1
17841.0	364.996941	8310.126768	3

8 out of 2790



Outcome

Further advice:

High CLV customers:
- Prioritize VIP services or personalized recommendation

Low CLV customers with high churn risk:
- Extend their lifecycle through coupons or win-back campaigns

THANK YOU