KOÇ UNIVERSITY

CSSM502- Advanced Data Analysis with Python

ML Based Customer Segmentation

Model Development Report

İzel Yazıcı - 77549
Egehan Coşgun – 40557

Table of Contents

Segmentation Methodology

Any customer segmentation, whether based on value or behavior attributes of customers, should be restricted to active customers since dormant and inactive customers constitute large chunks of similar groups. The quantitative method chosen for determining the activity window size is bounce-back analysis.

Step 1. Defining active, dormant, inactive and newcomer customers

Bounce-back Analysis

Based on bounce-back analysis, we decided to define customer status as active, dormant, inactive and newcomer. This analysis shows what proportion of customers' who shopped in particular year-month had no activity in next X months (churn rate) and what proportion of churn-tagged customers had a transaction in the succeeding X months. Since 9 months bounce-back rates are reasonably small and close to 12 months results, we decided to continue with 9 months transaction window for defining customer inactivity.

According to this definition and business decision on new-comers, we define macro-segments as follows:

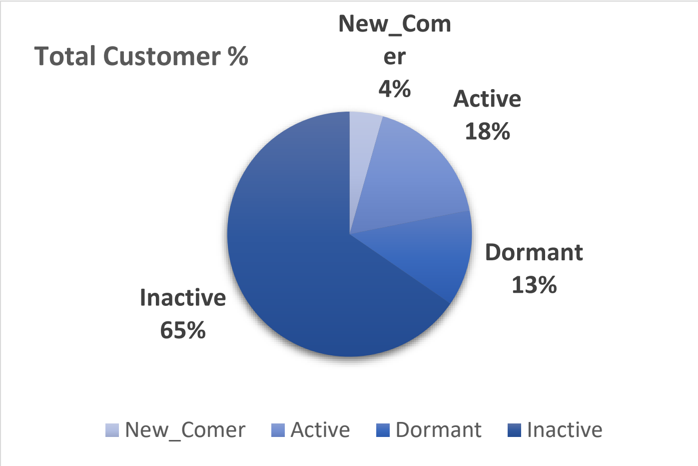
- If customer registration date is less than 120 days, it means Newcomer
- If customer last transaction is less than 270 days, it means Active
- If customer last transition is less than 730 days, it means Dormant
- If customer last transition is more than 730 days, it means Inactive

It is good to catch customer before going Inactive status. That's why we define customer status and help marketing to win back these customers.

Table 1. Churn and Bounceback analysis over different windows

		3M	6M	9M	12M
20170101	**Churn**	53%	35%	28%	25%
	BounceBack	35%	29%	25%	21%
20170102	**Churn**	50%	33%	28%	24%
	BounceBack	33%	27%	25%	22%
20170103	**Churn**	45%	32%	27%	24%
	BounceBack	30%	26%	23%	22%
20170104	**Churn**	45%	33%	28%	25%
	BounceBack	25%	25%	23%	21%
20170105	**Churn**	46%	35%	30%	26%
	BounceBack	24%	26%	24%	22%
20170106	**Churn**	51%	39%	34%	29%
	BounceBack	24%	24%	24%	21%
20170107	**Churn**	56%	43%	38%	33%
	BounceBack	23%	24%	24%	21%
20170108	**Churn**	59%	46%	39%	34%
	BounceBack	22%	26%	25%	21%
20170109	**Churn**	56%	44%	36%	32%

	BounceBack	22%	28%	25%	23%
20170110	**Churn**	47%	35%	26%	23%
	BounceBack	26%	35%	32%	33%
20170111	**Churn**	46%	33%	25%	21%
	BounceBack	29%	35%	33%	33%
20170112	**Churn**	44%	30%	23%	20%
	BounceBack	32%	33%	34%	34%



Segmentation Development

After the data analysis is completed, the necessary metrics that are planned to be used for the segmentation model and in the reporting data preparation process were completed .

The source table of metrics to be used in modeling was prepared in Python.

The diagram below expresses the development stages of the segmentation model in Python.

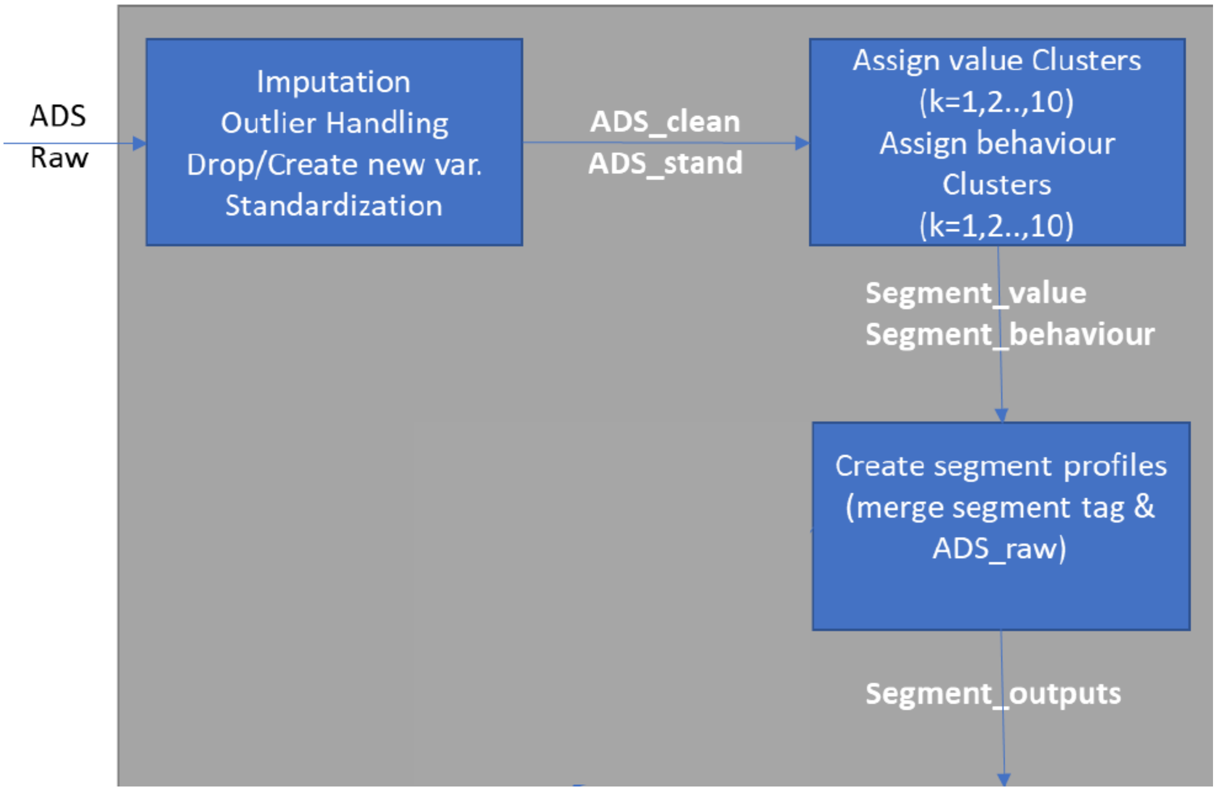


Figure 1. Clustering Model Development Stages for Segmentation in Python

Feature Generation

GM_Perc, Price Sensitivity, Frequecy and DaystoChurn dimensions were calculated.

- **Annual CLV** : Amount of GM we are expecting to earn from a customer in the next 12 months.

*CLV = Gross Margin Percentage * Basket Size * Shopping Frequency*

- **Frequency**: Mean interarrival time between client transaction dates
- **Price Sentivity**: Metric that measures the customer's price sensitivity. It is calculated as 1 minus full price ratio.
- **DaystoChurn**: The time remaining for the customer to churn is calculated as 270 days minus the last transaction date.

After calculating all the dimensions to be calculated, data preprocessing phase was started. The distributions of numeric variables were analyzed than outlier values were replaced with appropriate values.

Standardization

Values that should be between 0 and 1 were placed in this range, and other variables were replaced by not greater than the sum of the standard deviation and Q3 values. After the Outlier handling process was completed, **Reduce Price Ratio** was recalculated as **1-Full Price Ratio.**

GM_Perc has been compressed between the upper bound and the lower bound for use in this calculation in order not to break the CLV computation.

Metrics with different sizes were standardized between 0-1 to enter the clustering model. Thus, while creating clustering, the value of each variable will be evaluated within itself and the model has been made to work more smoothly.

Imputation

Finally, null values are imputed in a way that does not impair the significance of the data.

Customers' status were calculated based on the cleaned data.
- **Newcomer**: Customer who registered with the CRM system within the last 4 months.
- **Active**: Customer who has had at least 1 transaction for the last 9 months.
- **Churned**: Customer who has not had any transactions for the last 9 months.
- **Inactive**: Customer who has not had any transaction for the last 24 months. (These customers do not receive the data set in which the model is trained.)

Metrics with different sizes were standardized between 0-1 to enter the clustering model. Thus, while creating clustering, the value of each variable will be evaluated within itself and the model has been made to work more smoothly.

After the standardization process was done, the data pre-processing stage of the model was completed. This process is compiled under data_prep () function.
The input values of the data_prep () function are raw ADS (df_ADS_all) and brand_id, and the outputs are ADS_cleaned and ADS_stand. The output of this function, df_stand, was input to the clustering function and used as train data in the kmeans algorithm.

Clustering Model Train

The model was trained for each k value from 1 to 10 for each brand and elbow analysis was performed over the SSE Cluster value of the obtained models.

In the clustering model, the newcomer cluster with a customer tenure less than 4 months is not included in the training data set of the model. Segment outputs are added to the labeled as ClusterId= -1.

Behaviour Segmentation

At the begining, behaviour segmentations dimensions were **CustomerTenure, FrequencyMonth, PriceSentivity.**

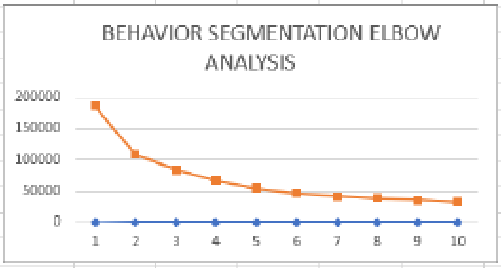
The effect of the **PointUsagePerc** dimension on the model was not found to be significant, and it was removed from the training data set of the model. Likewise, the **FullPriceRatio** dimension was replaced instead of the **PriceSensitivy** dimension because it is more effective in clustering themodel.

- **Peak Time:** Periods during which sales are significantly higher than average each year. (computed dynamically)
- **Peak Time Sales**: Proportion of customer spending during peak times in overall spending.
- **Distinct Department Count**: Number of distinct department groups customer has made purchase.
- **Customer Tenure**: Time elapsed after the customer registered with the CRM system

Elbow Analysis

Elbow analysis was made according to the SSE Cluster (inertia) value and the appropriate k value was decided for each brand.

ClusterVersion	SSECluster	SourceBrandKey	Marginal Delta as Perc
1	186510	1	42%
2	108154	1	22%
3	84198	1	20%
4	67278	1	19%
5	54730	1	15%
6	46286	1	10%
7	41798	1	10%
8	37738	1	8%
9	34903	1	7%
10	32508	1	



Segment profiles were created by evaluating the cluster outputs over KPIs, and in the light of this analysis and elbow analysis, it was decided how many different clusters each brand's behavior segment would be divided into.

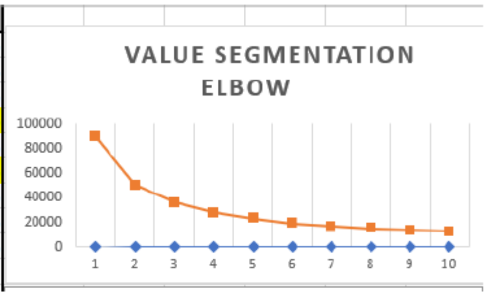
As a result of the elbow analysis, after deciding the most appropriate k value to be used in behavior segmentation for each brand, the trained model exported as a pickle file.

Value Segmentation

Value segmentations dimensions are **CustomerTenure, FrequencyMonth, GM_L24M_Perc** and **AVB_L24M.**

Segment profiles were created by evaluating the cluster outputs over KPIs, and in the light of this analysis and elbow analysis, it was decided how many different clusters each brand's value segment would be divided into.

ClusterVersion	SSECluster	SourceBrandKey	Marginal Delta as Perc
1	89604	1	44%
2	50349	1	28%
3	36322	1	23%
4	28101	1	18%
5	23110	1	17%
6	19208	1	12%
7	16861	1	11%
8	14940	1	9%
9	13607	1	6%
10	12790	1	



As a result of the elbow analysis, after deciding the most appropriate k value to be used in value segmentation for each brand, **the trained model** of each brand **is exported as a pickle file.**

Predict Clusters

The predict_cluster () function was created in order to load the pickle files obtained as a result of model development and run them monthly.
Before calling the **predict_cluster()** function, the value of k to be used in each brand's clustering algorithm is expressed with an if statement.