School of Computing

Predictive DecisionTree Classifier based on Customer RFM Segment

ANG KUO SHENG CLEMENT

(A0177726R)

Below are Field, Description and Filename of the retail dataset.

Field	*Description*	*Filename*
customer_Id	Unique identification number of a customer	Customer.csv
DOB	Date of Birth	Customer.csv
Gender	Gender of Customer	Customer.csv
city_code	City code Customer has registered	Customer.csv
transaction_id	Transactions Identifier	Transactions.csv
customer_Id	Customer Identifier	Transactions.csv
tran_date	Date of transaction performed	Transactions.csv
prod_subcat_code	Product Sub-Category	Transactions.csv
prod_cat_code	Product Category	Transactions.csv
Qty	Quantity purchased by customer	Transactions.csv
Store_type	Type of Stores	Transactions.csv
total_amt	Total Sales Amount	Transactions.csv
prod_cat_code	Product Category Code	prod_cat_info.csv
prod_cat	Product Category	prod_cat_info.csv
prod_subcat_code	Product Sub-Category	prod_cat_info.csv

(a) Describe the background of the dataset and identify an appropriate marketing problem that can be addressed with insights derived from the dataset.

With the retail market getting more and more competitive by the day, there has never been anything more important than the ability for optimizing service business processes when trying to satisfy the expectations of customers. Channelizing and managing data with the aim of working in favor of the customer as well as generating profits is very significant for survival. The dataset is a pseudo records of any sales transaction for various category of products from any online retailers.

Ideally, a retailer's customer data reflects the company's success in reaching and nurturing its customers. Retailers built reports summarizing customer behavior using metrics such as conversion

rate, average order value, recency of purchase and total amount spent in recent transactions. These measurements provided general insight into the behavioral tendencies of customers. It is important for a business to understand if their products are selling well. It is arguably more important to understand if their customers enjoyed the product they bought is enough to make a repeated purchase. One way to look at this is through customer retention rates, the average quantity purchased, and the average price by segmenting them into cohorts and analyze them using RFM Analysis with RFM Score formula based on quintiles.

Customer intelligence is the practice of determining and delivering data-driven insights into past and predicted future customer behavior. To be effective, customer intelligence must combine raw transactional and behavioral data to generate derived measures. In a nutshell, for big retail players all over the world, data analytics is applied more these days at all stages of the retail process – taking track of popular products that are emerging, forecasting of sales and future demand via predictive simulation, optimizing placements of products and offers through heat-mapping of customers and many others.

Retailers can also offer items that other customers who looked at similar items tend to purchase, fancy gift wrapping or some massively discounted products. Such incentives can make the overall value of the customer's basket go up significantly, along with their satisfaction.

Most marketers are already able to identify customer segmentation based on RFM analysis using any Business or Market Intelligence software, however, the challenge they face is finding predictive analytics solutions with prediction models that provide good precision and accuracy which is what we are tackling in this project.

The objective of this project is to build prediction models using DecisionTree Classification for customer segment type after performing the RFM analysis and Clustering. Each customer segment type could be map into customer retention or churn used as a target variable using to train the DecisionTree classification model.

(b) Describe and perform all the necessary steps to prepare the dataset for analysis.

1. To perform data cleaning (ie: Checking for any transactions_id with duplicate entries for removal & removal of data containing null values, negative values, invalid dates).

   Hence, any negative values in quantity or transactional figures are disregarded or converted into absolute figure to reduce the complexity for RFM analysis as well as predictive analysis using classification.

2. Perform Standard / Min-Max Scaling (ie: Data Normalization due to different statistical distribution, skewness, outliers found in certain variables).

Standard scaling was applied for data normalization on RFM features, before dataset is split into training/testing & applied for DecisionTree classification.

(c) Describe in detail how you would construct the analytic solution. Be sure to keep in mind the problem statement you have defined in Part (a) and identify the important components of your solution (e.g. approach(es)/technique(s) to be used, evaluation measure(s) etc.). In addition, provide the various settings/parameters that you may have applied in the construction of the analytic solution. Justify the settings applied.

1. RFM Analysis is applied using RFM Score formula based on 4 equal quintiles (25% group). RFM helps divide customers into various categories or clusters to identify customers who are more likely to respond to promotions and also for future personalization services. We can calculate and scoring the RFM for our customer used to identify specific clusters of customers (ie: BEST Customers, High-Spending New Customer, Lowest Spending Active Loyal, Loyal, Potential Churn – and thus allow any firms marketer to generate different marketing promotional campaign to increase customer retention, loyalty and customer lifetime value.

Calculate recency, frequency and monetary value using past user behaviour

20

	recency	frequency	monetary_value
customer_Id			
266783	457	5	14791.530
266784	815	3	5694.065
266785	658	8	35271.600
266788	366	4	6092.970
266794	1	12	28253.745
266799	93	4	9958.260
266803	1031	1	3984.630
266804	484	1	1588.990
266805	341	1	4623.320
266806	370	6	20229.235
266807	708	4	5816.720
266809	505	5	14995.955
266810	645	5	19846.905
266812	1177	2	1256.385
266813	317	1	1037.595

27

customer_Id	recency	frequency	monetary_value	r_quartile	f_quartile	m_quartile	RFMScore
266783	457	5	14791.530	2	2	2	222
266784	815	3	5694.065	4	4	3	443
266785	658	8	35271.600	3	1	1	311
266788	366	4	6092.970	1	3	3	133
266794	1	12	28253.745	1	1	1	111
266799	93	4	9958.260	1	3	2	132
266803	1031	1	3984.630	4	4	4	444
266804	484	1	1588.990	3	4	4	344
266805	341	1	4623.320	1	4	4	144
266806	370	6	20229.235	2	2	1	221

2. High RFM score means high LTV. Before building the machine learning model, we need to identify what is the type of this machine learning problem. A machine learning model can predict the $ value of the LTV. But here, we want LTV segments because it makes it more actionable and easy to communicate with other stakeholders. By applying K-means clustering, we can identify our existing LTV groups and build segments on top of it.
3. After we have RFM Scores for our customers, one of the ways is performing segmenting using K-means clustering. The number of clusters will be determined based on silhouette plot and elbow method for analysis from K-Means clustering. Based on the clustering result, marketing teams can conduct specific promotional campaign to different customer's clusters to retain more customers and maximize the store's profit.

   Below is a table with key RFM segments:

Segment	RFM	Description	Marketing
Best Customers	111	Bought most recently and most often, and spend the most	No price incentives, new products, and loyalty programs
Loyal Customers	X1X	Buy most frequently	Use R and M to further segment
Big Spenders	XX1	Spend the most	Market your most expensive products
Almost Lost	311	Haven't purchased for some time, but purchased frequently and spend the most	Aggressive price incentives
Lost Customers	411	Haven't purchased for some time, but purchased frequently and spend the most	Aggressive price incentives
Lost Cheap Customers	444	Last purchased long ago, purchased few, and spent little	Don't spend too much trying to re-acquire

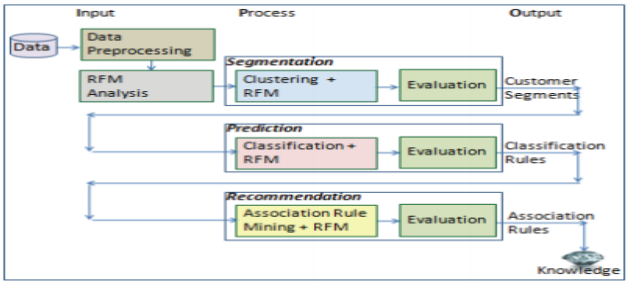
Best Customers – Communications with this group should make them feel valued and appreciated. These customers likely generate a disproportionately high percentage of overall revenues and thus focusing on keeping them happy should be a top priority. Further analyzing their individual preferences and affinities will provide additional opportunities for even more personalized messaging.

High-spending New Customers – It is always a good idea to carefully "incubate" all new customers, but because these new customers spent a lot on their first purchase, it's even more important. Like with the Best Customers group, it's important to make them feel valued and appreciated – and to give them terrific incentives to continue interacting with the brand.

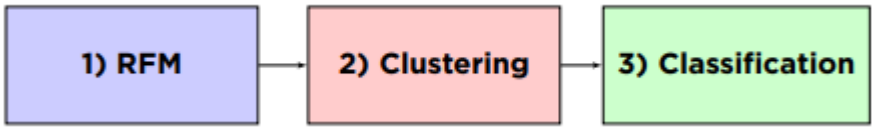
Lowest-Spending Active Loyal Customers – These repeat customers are active and loyal, but they are low spenders. Marketers should create campaigns for this group that make them feel valued, and incentivize them to increase their spend levels. As loyal customers, it often also pays to reward them with special offers if they spread the word about the brand to their friends, e.g., via social networks.

Churned Best Customers – These are valuable customers who stopped transacting a long time ago. While it's often challenging to re-engage churned customers, the high value of these customers makes it worthwhile trying. Like with the Best Customers group, it's important to communicate with them on the basis of their specific preferences, as known from earlier transaction data.

The important components of our approach/technique is that the predictor used for customer segmentation are behavioral related data, such as spending and consumption habits on category of product/service purchase instead of customer's psychographics, geographical, demographic which we have limited access to such information.



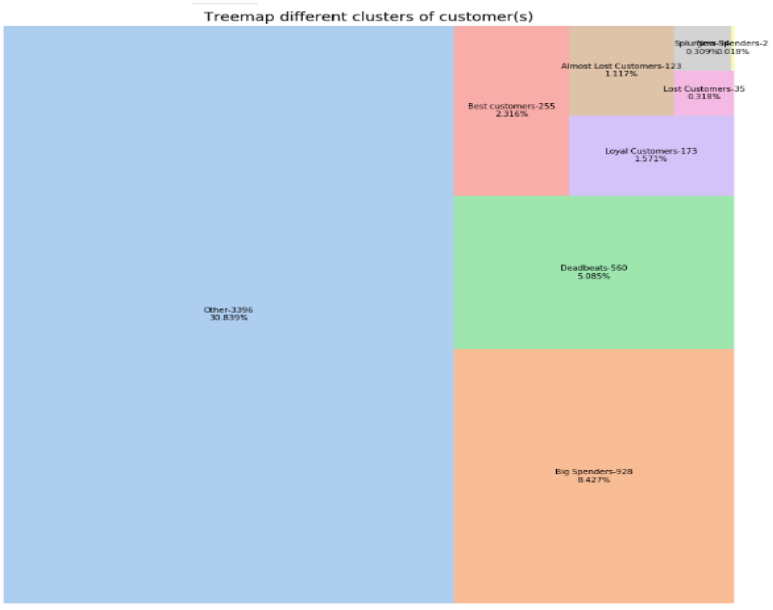
Once RFM and clustering is finished, below are the steps to build the training dataset for classification algorithms. Based on the normalized RFM score, we can apply classification algorithms such as multinomial Logistic Regression and Decision Trees to predict future customer behavior. This will be a multi-class classification problem with the number of classes corresponding to the number of clusters forming different classification of RFM LTV (ie: RFM Segment).



103

customer_Id	recency	frequency	monetary_value	r_quartile	f_quartile	m_quartile	RFMScore	Segment_Type
270831	215	12	53772.615	1	1	1	111	Best customers
271834	413	11	48425.520	2	1	1	211	NA
271862	357	11	44266.300	1	1	1	111	Best customers
267419	491	9	42951.350	3	1	1	311	NA
275252	344	12	42114.865	1	1	1	111	Best customers

- We will split our feature set and label (LTV) as X and y whereby X (RFM features) is used to predict y (Segment Types).
    - Having a pre-defined Training and Test dataset based on 70% training & 30% testing set is not a good choice because this way of partitioning the data leads to two major issues: (a) class imbalance and (b) sample representativeness issues. Class imbalance is known to affect the performance of many classifiers by introducing a bias towards the majority class of target variable (ie: such as Class 'Others'). As such, it is suggested to use stratified splitting on the target (dependent) variable to ensure that we don't train the classifier on imbalanced data that is observed in our target class. To overcome this class imbalance issue, this can be done by performing using Python StratifiedShuffleSplit(n_splits=5, test_size=0.2) to split dataset into 5 sets with test size=0.2.




```
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(segment_type_train, pred_gini_dtc)
```

Performance Measurement Metrics

Selecting the best metrics for evaluating the performance of a given classifier on a certain dataset is guided by a number of consideration including the class-balance and expected outcomes. One

particular performance measure may evaluate a classifier from a single perspective and often fail to measure others. Consequently, there is no unified metric to select measure the generalized performance of a classifier.

Hence, one of performance evaluation method is using classification report which provides the main classification metrics on a per-class basis reflecting its real performance using confusion matrix and classification reporting on precision, recall, F1, support.

	precision	recall	f1-score	support
Almost Lost Customers	1.00	1.00	1.00	86
Best customers	1.00	1.00	1.00	178
Big Spenders	1.00	1.00	1.00	650
Deadbeats	1.00	1.00	1.00	392
Lost Customers	1.00	1.00	1.00	25
Loyal Customers	1.00	1.00	1.00	121
New Spenders	1.00	1.00	1.00	1
Other	1.00	1.00	1.00	2376
Splurgers	1.00	1.00	1.00	24
accuracy			1.00	3853
macro avg	1.00	1.00	1.00	3853
weighted avg	1.00	1.00	1.00	3853

Tree Depth=5

a) Precision (tp / (tp + fp)) measures the ability of a classifier to identify only the correct instances for each class.

b) Recall (tp / (tp + fn) is the ability of a classifier to find all correct instances per class.

c) F1 score is a weighted harmonic mean of precision and recall normalized between 0 and 1. F score of 1 indicates a perfect balance as precision and the recall are inversely related. A high F1 score is useful where both high recall and precision is important.

d) Support is the number of actual occurrences of the class in the test data set. Imbalanced support in the training data may indicate the need for stratified sampling or rebalancing.

Another evaluation method is Confusion-matrix which yields the most ideal suite of metrics for evaluating the performance of a classification algorithm such as Logistic-regression or Decision-trees. It's typically used for binary classification problems but can be used for multi-label classification problems.

Accuracy with a binary classifier is measured as (TP+TN)/(TP+TN+FP+FN), but accuracy for a multiclass classifier is calculated as the average accuracy per class.

- Various settings/parameters that have been applied in the construction of the analytic solution

```
clf_gini = DecisionTreeClassifier(criterion='gini', splitter='best', max_depth=4,
random_state=0)
```

Tree depth=4 (Confusion Matrix)

	Other	Loyal Customers	Big Spenders	Almost Lost Customers	Best customers	Splurgers	Deadbeats	Lost Customers	New Spenders
Other	86	0	0	0	0	0	0	0	0
Loyal Customers	0	178	0	0	0	0	0	0	0
Big Spenders	0	0	650	0	0	0	0	0	0
Almost Lost Customers	0	0	0	392	0	0	0	0	0
Best customers	25	0	0	0	0	0	0	0	0
Splurgers	0	0	0	0	0	121	0	0	0
Deadbeats	0	0	0	0	0	0	1	0	0
Lost Customers	0	0	0	0	0	0	0	2376	0
New Spenders	0	0	0	0	0	0	0	0	24

```
clf_gini = DecisionTreeClassifier(criterion='gini', splitter='best', max_depth=5,
random_state=0)
```

Tree depth=5 (Confusion Matrix)

	Big Spenders	Other	Best customers	Almost Lost Customers	Deadbeats	Loyal Customers	Splurgers	Lost Customers	New Spenders
Big Spenders	86	0	0	0	0	0	0	0	0
Other	0	178	0	0	0	0	0	0	0
Best customers	0	0	650	0	0	0	0	0	0
Almost Lost Customers	0	0	0	392	0	0	0	0	0
Deadbeats	0	0	0	0	25	0	0	0	0
Loyal Customers	0	0	0	0	0	121	0	0	0
Splurgers	0	0	0	0	0	0	1	0	0
Lost Customers	0	0	0	0	0	0	0	2376	0
New Spenders	0	0	0	0	0	0	0	0	24

Classification Performance Report

```
print(" Classification Report for DecisionTree Classifier")
print(classification_report(segment_type_train, pred_gini_dtc))
```

```
                         precision    recall  f1-score   support

  Almost Lost Customers       1.00      1.00      1.00        86
         Best customers       1.00      1.00      1.00       178
           Big Spenders       1.00      1.00      1.00       650
              Deadbeats       1.00      1.00      1.00       392
         Lost Customers       1.00      1.00      1.00        25
        Loyal Customers       1.00      1.00      1.00       121
           New Spenders       1.00      1.00      1.00         1
                  Other       1.00      1.00      1.00      2376
              Splurgers       1.00      1.00      1.00        24

               accuracy                           1.00      3853
              macro avg       1.00      1.00      1.00      3853
           weighted avg       1.00      1.00      1.00      3853
```

- Justify the settings applied

Criterion = 'gini': is intended for continuous attributes & to minimize misclassification. Criterion Parameter is used to define different impurity measurement. Unlike Entropy which is intended for

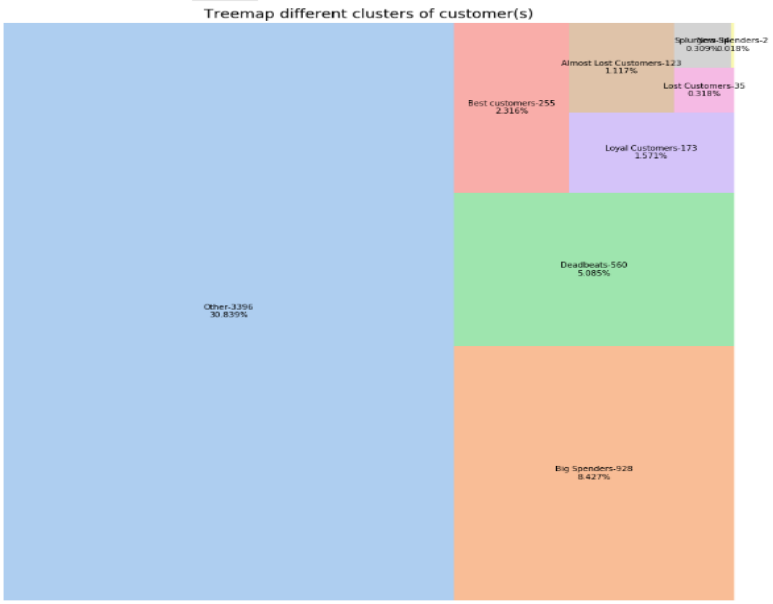
attributes that occur in classes & is for exploratory analysis. As per parsimony principal Gini outperform entropy, But as of computation ease (log is obvious has more computations involved rather that plain multiplication at processor/Machine level). Entropy takes slightly more computation time than Gini Index because of the log calculation, maybe that's why Gini Index is the option. But entropy definitely has an edge in some data cases involving high imbalance

Splitter='best: best split is when separating the classes accurately based on that feature. The strategy is "best" used to choose the best split at each node instead of "random" to choose the best random split

Max_Depth: If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples. The depth of the tree will influence the complexity of predictive model because there'll be more splits and it captures more information about the data and this is one of the root causes of over-fitting in decision trees as model will fit perfectly for the training data and will not be able to generalize well on test set. With low depth, the model will under-fit. To find the best value is subjective because over-fitting and under-fitting are very subjective to a dataset. I experiment with depth=4 and 5 and make observation on the confusion matrix whether there's any mis-classification. Overall higher accuracy is also preferred choice. Hence, the decision was for depth tree used is 5.

(d) Evaluate and comment on the results/outputs you have derived from your analytic solution. Justify how the results/outputs address the problem statement defined in Part (a) and comment whether the insight(s) derived is (are) useful in solving the problem.

From Treemap for different RFM segments, it shows ratio of best customers and loyal customers are at low proportion of about 4% and big spenders of 8%, and large proportion of segment is "others", we will really need to identify which features of RFM play an important role to predict the different clusters of customer(s). The Treemap showing different segments of customers with different proportion help us to refine different marketing strategies to improve customer retention and decrease churn rate.



For each of RFM segments, it require different marketing strategies how to attract & retain the different customer base on RFM analysis to address the problem in part (a). For 'best customer' segments, these group that have bought recently, buy often and spend the most. It's likely that they will continue to do so & can consider marketing to them without price incentives to preserve profit margin. For new products launch, these customers should be personalized via short message text or email marketing whereby their membership should qualify them for rewards perk or loyalty discounts.

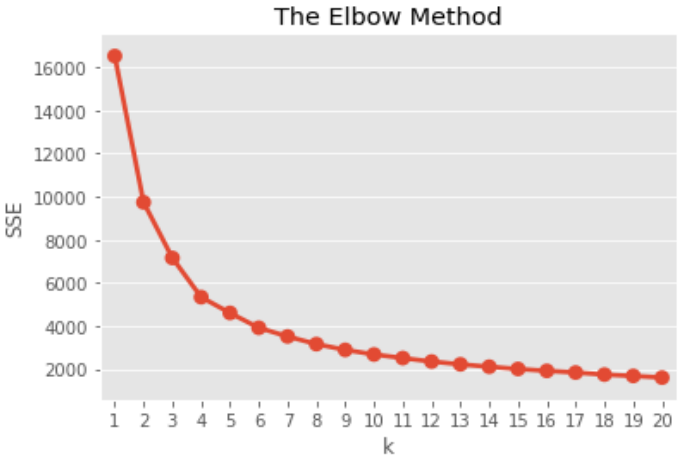
Meanwhile, the "Loyal" segment are those buy often, but don't spend very much would require marketing campaign surveys & personalization in ways to increase their spending.

The "almost Lost" or "Lost" segment which can form the churn model as "YES" and the rest as "No", we definitely would consider RFM features valuable to develop for churn modelling in predicting these group/cluster which used to buy frequently or spent a lot at one point, but they've stopped.

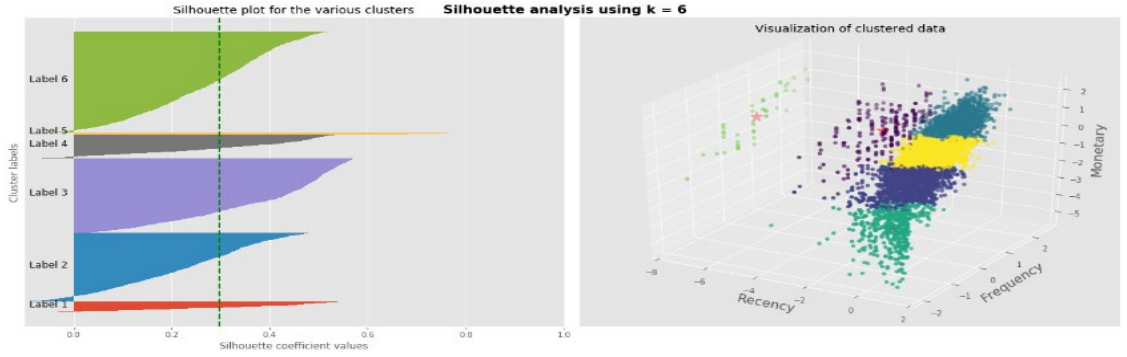
From this TreeMap data visualization, it should aid marketing & business development team in refining their marketing strategies to review their efforts & improvements progress.

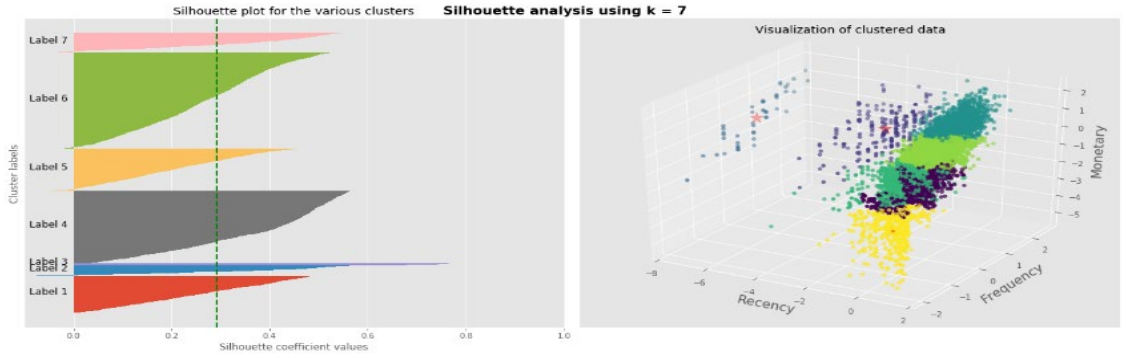
K-means clustering

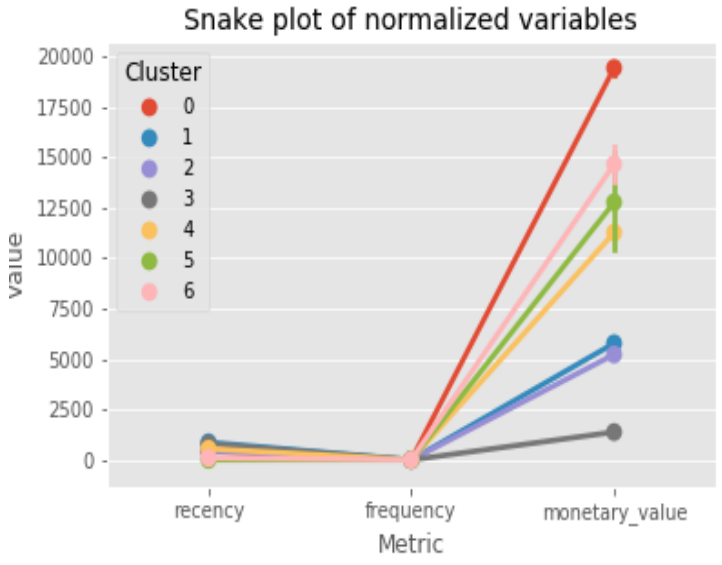
RFM log transformation and normalization was done before applying to K-means clustering, to generate the Elbow plot and Silhouette analysis. The optimum number of clusters is estimated to be 6.



The optimum number of clusters seems to be 6 even with the silhouette analysis







We can focus on Cluster 0 to improve revenue

Based on this snake plot diagram, we can determine which cluster will help to improve the most revenue. Therefore, the cluster 0 will be paid attention to improve revenue as this cluster has the highest value in monetary_vaue.

- Metric Measurement (Classification Report)

```
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix

print(classification_report(segment_type_train, pred_gini_dtc))

cm = confusion_matrix(segment_type_train, pred_gini_dtc)

# Instantiate the confusion matrix DataFrame with index and columns

cm_df = pd.DataFrame(cm, index = segment_type_train.unique(), columns=
segment_type_train.unique())

print(cm_df)
```

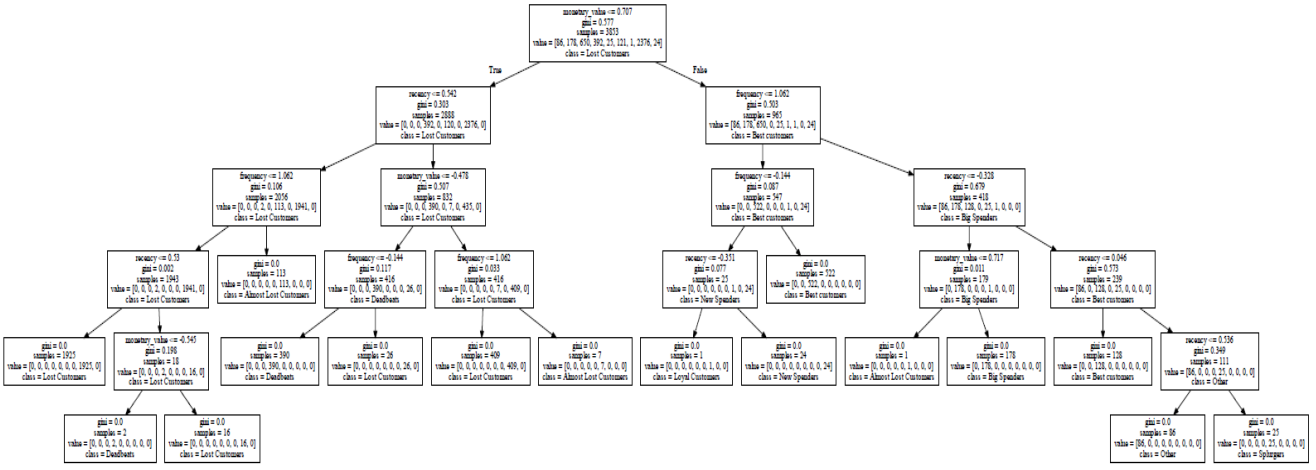
Based on the tree depth of 5, below is the results of the classification report for DecisionTreeClassifer model with Gini criterion and best splitter parameters settings as mentioned earlier. The results of Precision, Recall and f1-score determine the choice of the tree depth & justify for the parameters settings made in DecisionTree Classifier.

```
                        precision    recall  f1-score   support

  Almost Lost Customers       1.00      1.00      1.00        86
         Best customers       1.00      1.00      1.00       178
           Big Spenders       1.00      1.00      1.00       650
              Deadbeats       1.00      1.00      1.00       392
         Lost Customers       1.00      1.00      1.00        25
        Loyal Customers       1.00      1.00      1.00       121
           New Spenders       1.00      1.00      1.00         1
                  Other       1.00      1.00      1.00      2376
              Splurgers       1.00      1.00      1.00        24

               accuracy                           1.00      3853
              macro avg       1.00      1.00      1.00      3853
           weighted avg       1.00      1.00      1.00      3853
```

Using the feature importance from the DecisionTree, "monetary_value" contributes 50% in predicting the customer segment. However, if a feature has a low feature_importance, it doesn't mean that this feature is uninformative. It only means that this feature was not picked by the tree, likely because recency and frequency encodes the same information

Below is the DecisionTree Classifier diagram generated, it aids human eyes in the visualization to determine whether any nodes that contains little information can be pruned which is known as post-pruning.

We see the recency <=0.542 & monetary_value <=0.707, it can be classified in the class or segment type "Lost Customers". In contrast, recency >=0.542 and monetary_value > 0.707 can be classified as "Best Customers"



We can observe pure leaf since gini=0 when tree depth=5.

(e) Discuss the limitation(s) on the selected approach/solution and suggest how the limitation(s) can be mitigated/addressed.

Limitation:

Reliability of the information in the decision tree depends on feeding the precise internal and external information at the onset. Even a small change in input data can at times, cause large

changes in the tree. Changing variables, excluding duplication information, or altering the sequence midway can lead to major changes and might possibly require redrawing the tree.

1. A small change in the data can cause a large change in the structure of the decision tree causing instability.
2. For a Decision tree sometimes calculation can go far more complex compared to other algorithms.
3. Decision tree often involves higher time to train the model.
4. Decision tree training is relatively expensive as complexity and time taken is more.

Decision tree complexity has a crucial effect on its accuracy, precision and it is influenced by any of these parameters with stopping criteria such as splitting method criterion (ie: entropy or gini), depth of tree employed. Stopping rules must be applied when building a decision tree to prevent the model from becoming overly complex.

Common parameters used in stopping rules include: (a) the minimum number of records in a leaf; (b) the minimum number of records in a node prior to splitting; and (c) the depth (ie: number of steps) of any leaf from the root node. Stopping parameters must be selected based on the goal of the analysis and the characteristics of the dataset. Typically, the tree complexity is measured by one of the following metrics: the total number of nodes, total number of leaves, tree depth and number of attributes used.

Mitigation

1. Since stopping rules do not work well, an alternative way to build a decision tree model is to grow a large tree first, and then prune it to optimal size by removing nodes that provide less additional information. A common method of selecting the best possible sub-tree from several candidates is to consider the proportion of records with error prediction (i. e. , the proportion in which the predicted occurrence of the target is incorrect).

2. Other methods of selecting the best alternative is to use a validation dataset (ie: dividing the sample in two and testing the model developed on the training dataset on the validation dataset), or, for small samples, cross-validation (ie: dividing the sample in 10 groups or 'folds', and testing the model developed from 9 folds on the 10th fold, repeated for all ten combinations, and averaging the rates or erroneous predictions).