

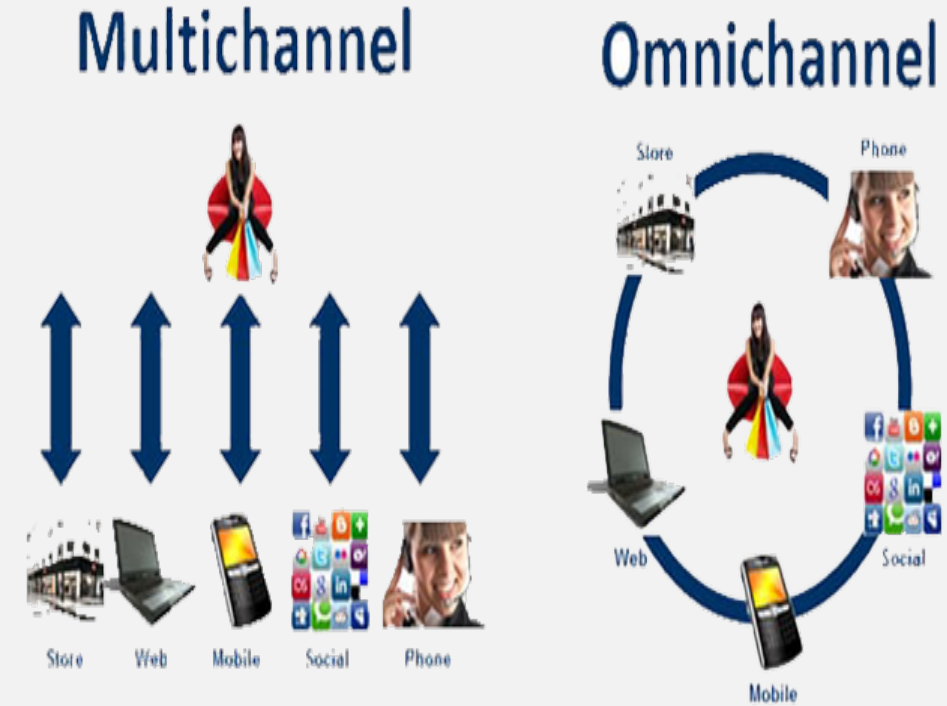


# Predictive Classifier on Customer RFM Segment

**By** ANG Kuo Sheng Clement

# PROBLEM STATEMENT

- ecommerce market getting more and more competitive with many new entrants globally & locally, no barriers to entry among competitors rivalry
- making it challenging to achieve customer retention via multichannel, omni-channel content and marketing strategies
- Be it offline brick-n-mortar store or e-commerce store using live-streaming app (ie: Alibaba) to connect social aspect of life to e-commerce, there's constant need for identifying Customer Lifestyle Value into segments to approach with targeted marketing strategies & campaigns



# Objective

- Used as Customer Relationship Management to identify which marketing campaign or customer rewards scheme works for respective Customer Segment.
- Based on Customer RFM Segment Value, develop predictive model of Customer Life-Time Value (CLTV)
- To train DecisionTree classifier with optimal tree depth. Model suitability is evaluated via F1 score as well as Accuracy and Specificity.

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

# DATASET

Dataset is like any typical sales transactions from retail e-commerce store

Necessary steps to prepare the dataset before performing any descriptive, prescriptive and predictive 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).

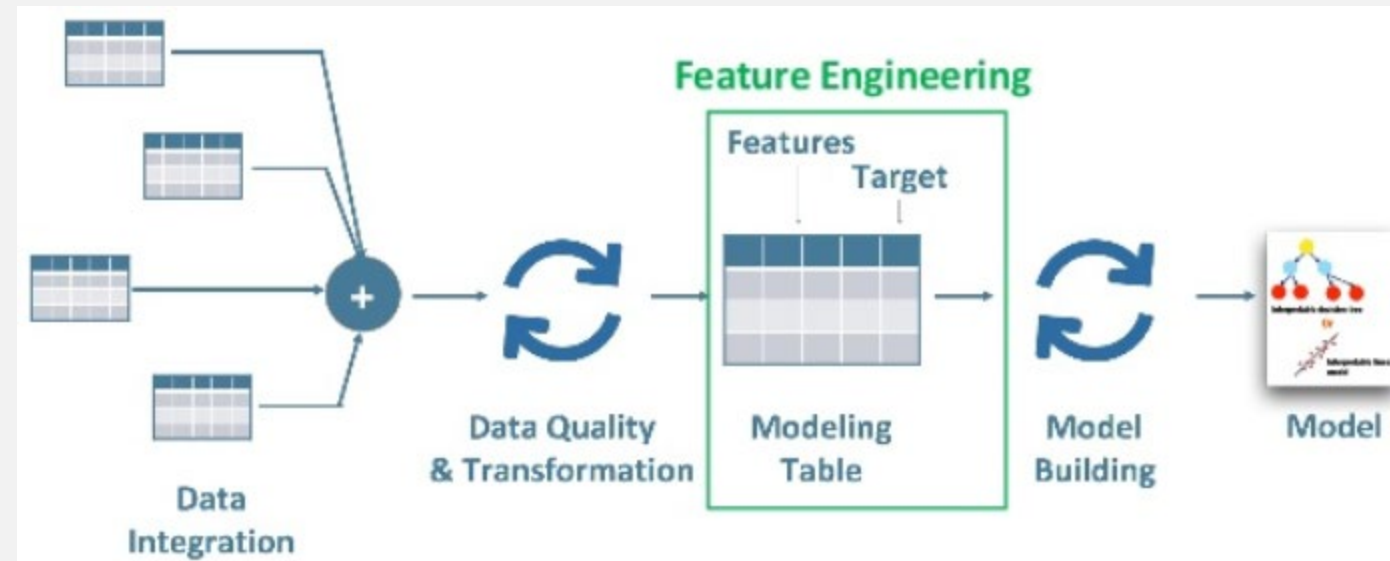
Hence, any negative values in quantity or transactional figures are disregarded or converted into absolute figure to reduce the complexity for RFM analysis as well as predictive analysis using classification.

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

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

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

# DATASET FEATURES



Features selection of dataset used for analysis:

1. Consumer's spending behavioral related data, such as spending and consumption habits on category of product/service purchase for all the transactions
2. Store Type is useful for tracking customer engagement in omni-channel distribution or multi-channel

# Descriptive Analysis

```
cs_freq_summary =  
cs_df.groupby(['customer_Id'])['transaction_id'].aggregate({'freq_transaction_count': 'count'})  
  
cs_freq_summary['Repeat_one_time'] = np.where((cs_freq_summary.freq_transaction_count > 1),  
                                              'repeat', 'one-time')  
  
print(cs_freq_summary['Repeat_one_time'])  
plt.style.use('ggplot')
```

```
media_df = cs_freq_summary  
media_per_user_group = media_df.groupby(['Repeat_one_time'])['Repeat_one_time'].count().nlargest(2)  
media_per_user_group.plot(kind='pie', colors = ['teal', 'gold'], fontsize=12, autopct='%1.1f%%',  
                           startangle=90, pctdistance=0.85)  
plt.show()
```

Proportion of transactions with repeat buy (recurring sales)



- there's a small fraction 8% of customers who has only perform a single transaction & 92% with repeat buy transaction, depicts the opportunity to address on F (frequency) within RFM analysis for CLV (Customer Lifetime value).

```
# customer & transaction counts  
cs_transac_prod_cat =  
cs_df.groupby(['customer_Id', 'transaction_id'])['prod_cat', 'Store_type'].  
aggregate('count').reset_index().sort_values('customer_Id',  
                                               ascending=True).head(20)
```

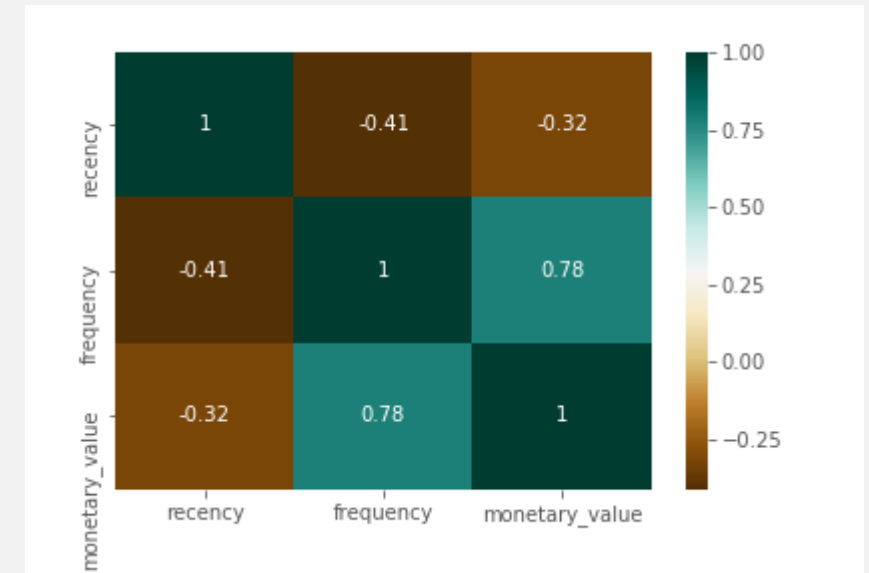
Customer transaction profile

92

	customer_id	transaction_id	prod_cat	Store_type
0	266783	8410316370	1	1
1	266783	16999552161	1	1
2	266783	25890929042	2	2
3	266783	98477711300	1	1
4	266784	26928161256	1	1
5	266784	36310127403	1	1
6	266784	54234600611	1	1
13	266785	96176911576	2	2
12	266785	94925617839	1	1
10	266785	79527990288	1	1
11	266785	89882144571	1	1
8	266785	62414620900	1	1
7	266785	17960226367	1	1

# Descriptive Analysis

- Frequency and monetary value are positively correlated with each other implying an increase in frequency implies increase in monetary value
- 
- Frequency and Recency are negatively correlated with each other implying an increase in frequency implies decrease in monetary value



# Prescriptive Analysis

RFM Analysis is applied using RFM Score formula based on 4 equal quintiles (25% group), which divide customers into various segments or clusters, also known as Customer Segmentation.

Purpose of RFM Analysis:

- identify customers reaction & respond to promotions and also for future personalization services
- Improve marketing performance by making campaigns relevant to customers, thus increasing response rate and sales revenue
- allow firms marketer to generate different marketing strategies or promotional campaign accordingly to increase customer retention, loyalty and customer lifetime value

Type of Customer Segments/Clusters:

(ie: BEST Customers, High-Spending New Customer, Lowest Spending Active Loyal, Loyal, Potential Churn )

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

RFM Segment Table using past purchase transaction history

	recency	frequency	monetary_value	r_quartile	f_quartile	m_quartile	RFMScore
customer_id							
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



# Prescriptive Analysis

This treemap also show the class imbalance of RFM segment that needs to be handled differently instead of using predefined partitioning.

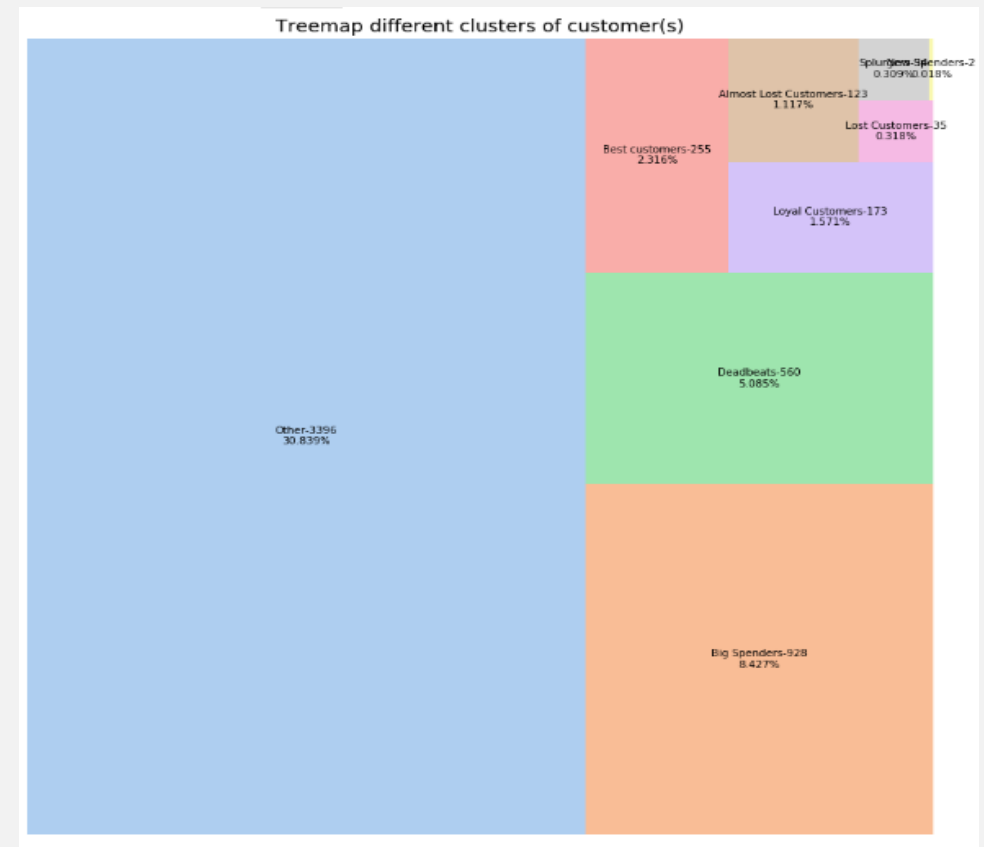
Pre-defined partitioning (80-20%) affects many classifiers model's performance through the lack of opportunity for the algorithm to learn due to sample representative having class imbalance, bias or skewed issue. Class imbalance is known to affect the performance of many classifiers by introducing a bias towards the majority class of target variable such as "Others".

Hence, random sampling is recommended to achieve independence and also a smooth generation of samples without any bias on the training set. Alternatively, splitting on the target (dependent) variable to ensure that we don't train the classifier on imbalanced data.

```
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3,  
random_state=123, stratify=y)
```

```
skf = StratifiedKFold(n_splits=10,random_state=1).split(X_train, y_train)
```

Treemap Generated from dataset using Python

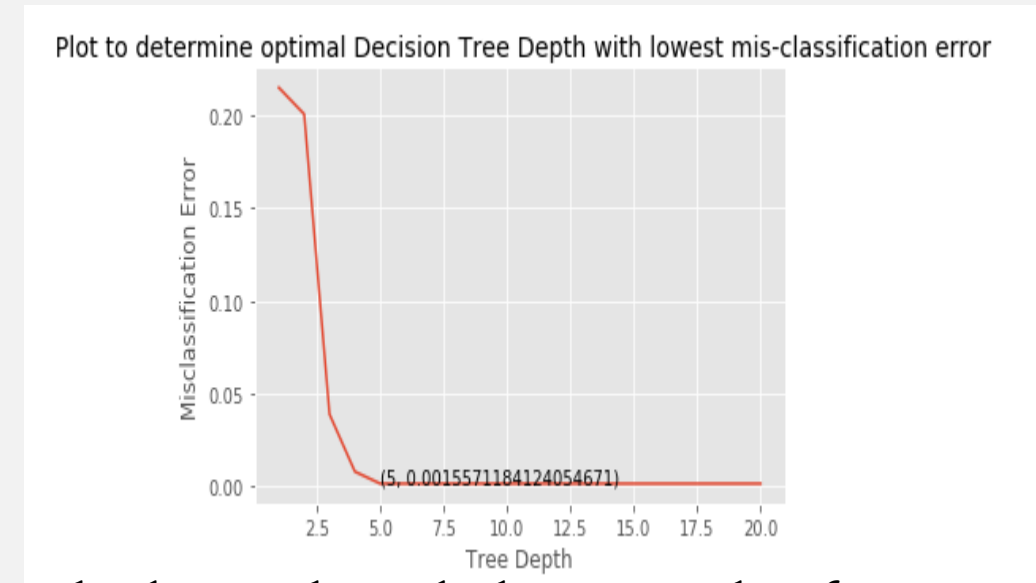


# Model

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

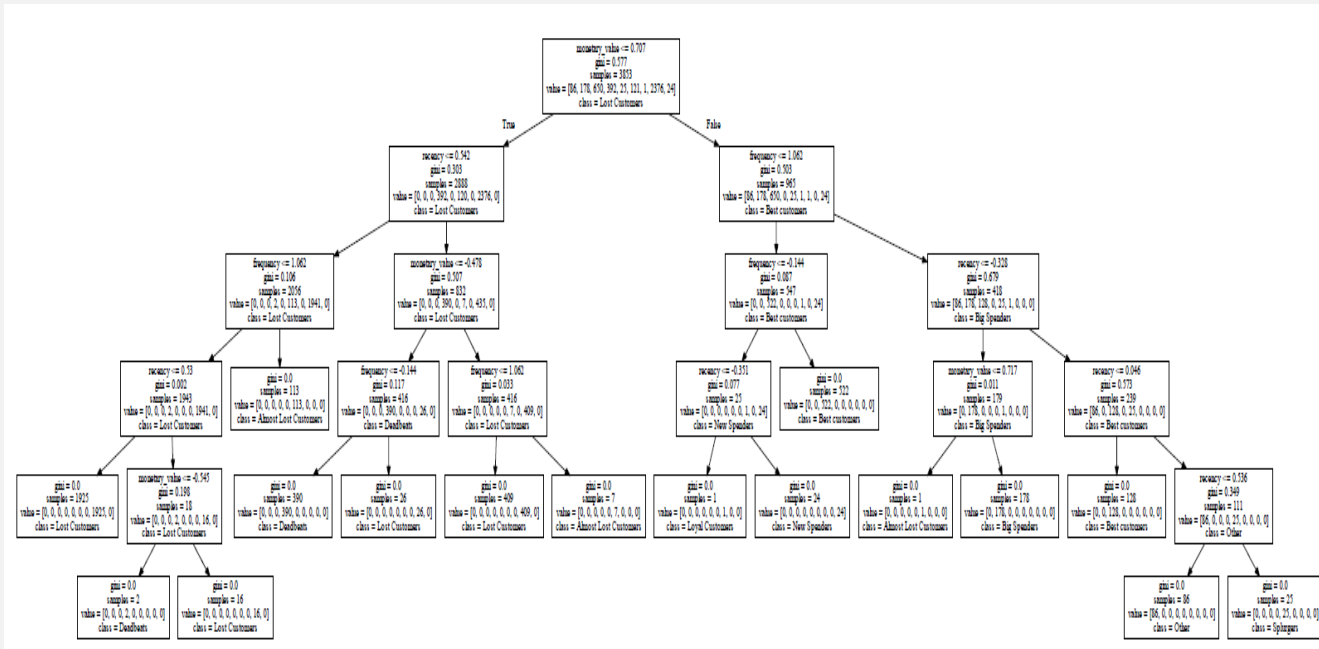
```
clf_gini =  
DecisionTreeClassifier(criterion='gini',  
splitter='best', max_depth=4, random_state=0)
```

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



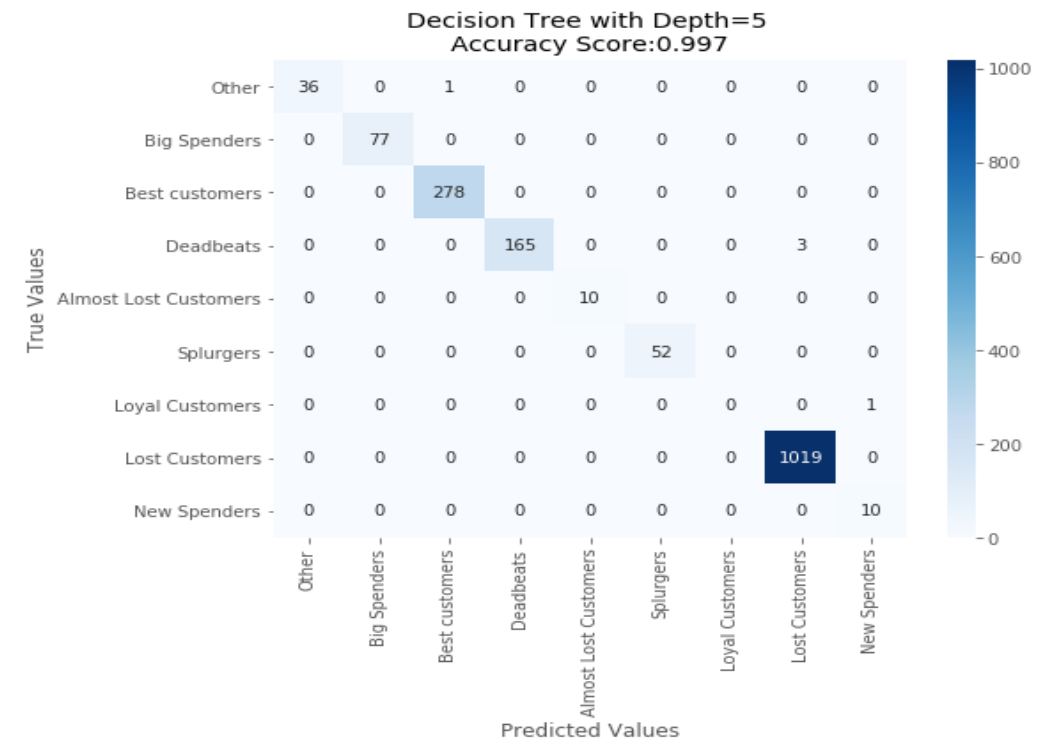
Since tree depth=5 produces the lowest misclassification error, it will be used as the predictive model

# MODEL EVALUATION



With stratified sampling & when tree depth=5, we can observe there is pure leaf or node with no impurity.

Confusion Matrix of DecisionTree with Depth=5



From this confusion matrix, it shows misclassification error exist